Karol BASINSKI, Bartlomiej UFNALSKI, Lech M. GRZESIAK

Warsaw University of Technology, Institute of Control and Industrial Electronics

doi:10.15199/48.2017.02.40

Particle swarm based repetitive spline compensator for servo drives

Abstract. *In this paper the particle swarm based repetitive spline compensator (PSBRSC), a new method of repetitive compensator implementation, is investigated. The proposed approach employs the particle swarm optimizer (PSO) to solve a dynamic optimization problem (DOP) related to the control task in a servo drive with a permanent magnet synchronous machine (PMSM) in online mode. The first novelty reported here is to use cubic spline interpolation to calculate the samples of PSBRSC signal that are located between the samples taken directly from the optimizer. Also the responsiveness of the repetitive controller is improved thanks to the introduction of the evaporation rate growth mechanism.*

Streszczenie. *W artykule przedstawiono kompensator splajnowo-rojowy (ang. particle swarm based repetitive spline compensator), nową metodę realizacji kompensacji w procesach powtarzalnych. Zaproponowany układ wykorzystuje metodę roju cząstek do rozwiązywania w czasie rzeczywistym zagadnienia optymalizacji dynamicznej związanego z kształtowaniem sygnału modyfikującego uchyb regulacji w serwonapędzie z silnikiem synchronicznym z magnesami trwałymi (PMSM). Pierwszą nowością przedstawioną w artykule jest wykorzystanie interpolacji splajnowej trzeciego rzędu do wyznaczenia próbek sygnału wyjściowego kompensatora znajdujących się pomiędzy próbkami pochodzącymi bezpośrednio z optymalizatora. Ponadto szybkość reakcji kompensatora została poprawiona dzięki wprowadzeniu mechanizmu wzrostu współczynnika zapominania.* ***(Kompenstor splajnowo-rojowy do procesów powtarzalnych dedykowany dla serwonapędów)***

Keywords: repetitive process control, particle swarm optimization, dynamic optimization problem, cubic spline interpolation, servo drive
Słowa kluczowe: sterowanie procesami powtarzalnymi, optymalizacja metodą roju cząstek, zagadnienie optymalizacji dynamicznej, interpolacja splajnowa, serwonapęd

Introduction

Automatic control systems are often based on PID controllers [1, 2, 3]. This solution has several advantages, such as low computational complexity, no requirement to calculate the model of plant dynamics in real time and the fact that the controlled variable is the only signal required to be measured. The disadvantages may include some knowledge about the process being ignored, such as its repetitiveness. Servo drives are often used in repetitive processes where reference trajectory is supposed to be repetitive throughout the process. This property can be used to improve the effectiveness of the control scheme. One of the control schemes that takes repeatability of the process into account is iterative learning control (ILC) [4, 5, 6, 7, 8, 9, 10, 11, 12, 13]. The operating principle of the classic ILC is described by the following equation:

$$u\left(p,k\right)=u\left(p,k-1\right)+k_{\mathrm{RC}}e\left(p,k-1\right), \tag{1}$$

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 and p is the time index along the pass ($1 \leq p \leq N$, where N is the pass length).

Another control method that uses the information about the repetitiveness of a process to reduce control error is repetitive control (RC). A historical difference between ILC and RC lies in the fact that in the classic ILC the initial state of each pass can be reset [7]. This is because ILC was originally developed for batch processes, whereas RC for continuous repetitive processes. Nevertheless, it is quite common that a particular control technique can be applied equally successfully in either of the above process types. This happens, e.g., in the case of DOP-based repetitive controllers.

The main idea of the proposed control scheme is to add the output signal of PSBRSC (u_{spline}) to the control error, which is shown in Fig. 1 and which results in the change of

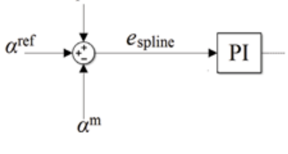


Fig. 1. Idea of proposed control scheme, modification of PI controller input

PI controller input from

$$e=\alpha^{\mathrm{ref}}-\alpha^{\mathrm{m}} \tag{2}$$

to

$$e_{\mathrm{spline}}=\alpha^{\mathrm{ref}}-\alpha^{\mathrm{m}}+u_{\mathrm{spline}} \tag{3}$$

where e and e_{spline} denotes PI input, α^{ref} is the reference signal, α^{m} is the measured signal and u_{spline} is the spline compensator output.

This paper presents simulational results of PSBRSC applied to a servo drive with a permanent magnet synchronous machine.

Particle swarm based repetitive spline compensator

In the plug-in direct particle swarm repetitive controller (PDPSRC) [14] to find the best control signal, the swarm itself is a repetitive controller cooperating with the controller shaping the signal behaviour along the pass. The initial single-swarm controller (PDPSRC) uses all the values of control error from the previous pass to update current control signal sample. Moreover, this algorithm is global in terms of the component of the objective function used in the definition of the update law, namely the mean squared control error calculated for the entire pass. However, to improve the convergence rate of the swarm without a significant loss of output voltage quality, a multi-swarm approach has been proposed [15]. It makes the PDPSRC algorithm non-global because the reference signal period is segmented into shorter intervals and the control signal is optimized in each interval independently by separate swarms. In this approach we distribute the dynamic optimization problem (DOP) among fewer dimensional swarms.

In this paper, the particle swarm based repetitive spline compensator based on the PDPSRC algorithm is presented. To achieve global optimization and less dimensional optimization problem than the number of samples of the measured signal, only every hundredth sample of u_{spline} signal comes directly from the optimizer, which is demonstrated in Fig. 2. The remaining u_{spline} signal samples are calculated using cubic spline interpolation [16], which is shown in Fig. 3. In PSBRSC spline interpolation replaces multi-swarm approach from PDPSRC to reduce swarm dimensionality. Furthermore we use only one swarm, which makes this

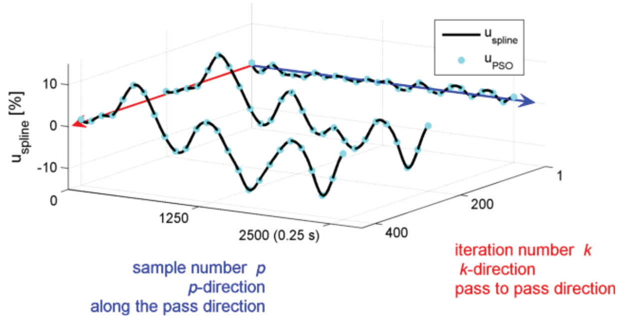


Fig. 2. Idea of control in the p and k-direction using the base value of $\alpha^{\mathrm{ref}}_{\mathrm{max}}$

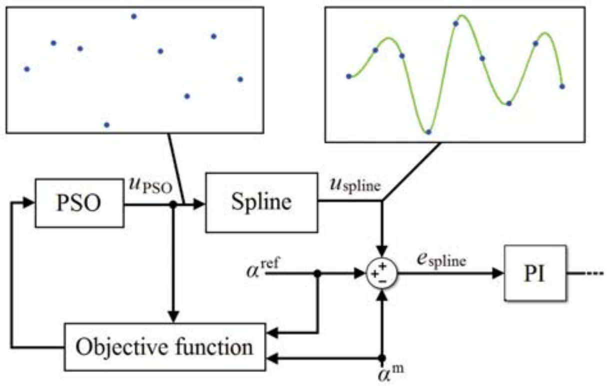


Fig. 3. Idea of particle swarm based spline compensator

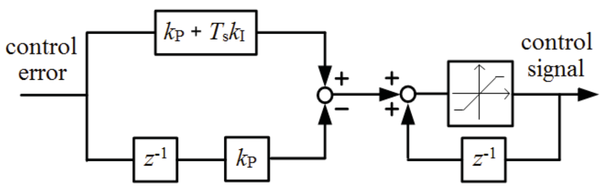


Fig. 4. Anti-windup PI controller block diagram

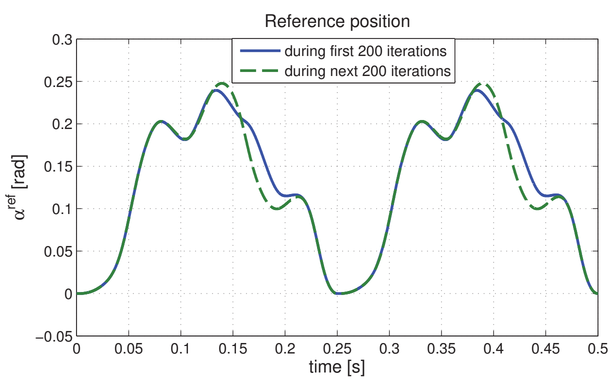


Fig. 5. Reference positions

Table 1. Parameters of plant and swarm

Parameter name	Value/Description
Stator resistance	0.266 Ω
Stator inductance d-axis	3.488 μH
Stator inductance q-axis	3.488 μH
Rated speed	330 r/min
Rated torque	145.67 N m
Moment of inertia	0.4224 kg m^2
Number of pole pairs	17
Sampling time	T_s=100 μs (N=2500 points per pass)
Measurement noise	2 % of 30 A (band-limited white noise with 98 % of its samples within the range)
System noise induced at the modulator stage	1 % of 170 V (band-limited white noise with 98 % of its samples within the range)
Swarm size	15
Particle dimension	25
Base value of evaporation rate	1.04

algorithm global. The PSO objective is to minimize in real time:

$$\mathcal{J}(k) = \mathcal{J}_0 + \underbrace{\sum_{p=1}^{N}\left(\alpha^{\mathrm{ref}}(p)-\alpha^{\mathrm{m}}(p,k)\right)^2}_{\text{penalty for control error}} + \underbrace{\beta \sum_{p_{\mathrm{PSO}}=2}^{N_{\mathrm{PSO}}}\left(u_{\mathrm{PSO}}(p_{\mathrm{PSO}},k)-u_{\mathrm{PSO}}(p_{\mathrm{PSO}}-1,k)\right)^2}_{\text{penalty for control signal dynamics}}, \tag{4}$$

or

$$\mathcal{J}(k) = \mathcal{J}_0 + \underbrace{\sum_{p=1}^{N}\left(\alpha^{\mathrm{ref}}(p)-\alpha^{\mathrm{m}}(p,k)\right)^2}_{\text{penalty for control error}} + \underbrace{\beta \sum_{p=2}^{N}\left(u_{\mathrm{spline}}(p,k)-u_{\mathrm{spline}}(p-1,k)\right)^2}_{\text{penalty for control signal dynamics}}, \tag{5}$$

where β denotes the penalty factor (determined by guessing and checking), N is the number of samples, N_{PSO} denotes the number of PSO particle dimensions, k denotes iteration number in the pass to pass direction, α^{ref} is the reference signal, α^{m} denotes the measured signal, u_{spline} is the PSBRSC output signal, u_{PSO} represents the particle swarm otimizer output. The positive constant $\mathcal{J}_0$ in (4) is introduced to ensure a positive definiteness of the performance index, which is crucial for the knowledge evaporation mechanism [15, 14] that has to be used if the optimization task at hand is of the DOP type (the optimal solution may vary during optimization). We employ PSO to solve the DOP type because tools are subject to wear during ongoing work and repetitive trajectory and load may be changed after some iterations. These reasons render an optimal solution obsolete. The minimization of $\mathcal{J}$ is done by particles the speed and position update rules of which are as follows:

$$\mathbf{v}_j(i+1) = c_1\mathbf{v}_j(i) + c_2 r^{\mathrm{pbest}}\delta_p\left(\mathbf{q}_j^{\mathrm{pbest}} - \mathbf{q}_j(i)\right) + c_3 r^{\mathrm{gbest}}\delta_p\left(\mathbf{q}^{\mathrm{gbest}} - \mathbf{q}_j(i)\right) \tag{6}$$

$$\mathbf{q}_j(i+1) = \mathbf{q}_j(i) + \min\{\max\{-v_{\mathrm{clmp}}, \mathbf{v}_j(i+1)\}, v_{\mathrm{clmp}}\}, \tag{7}$$

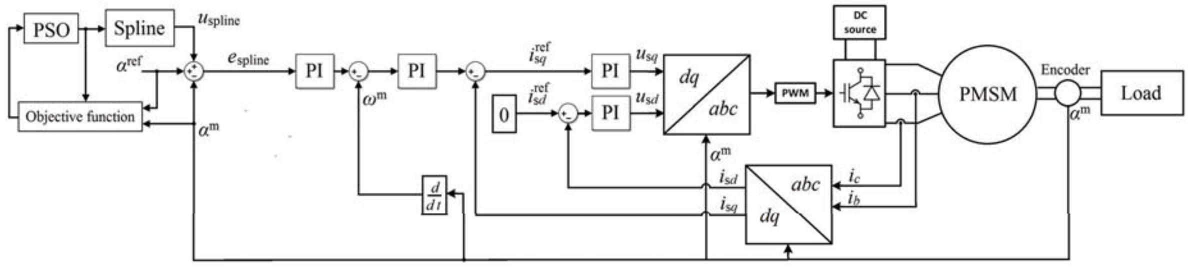
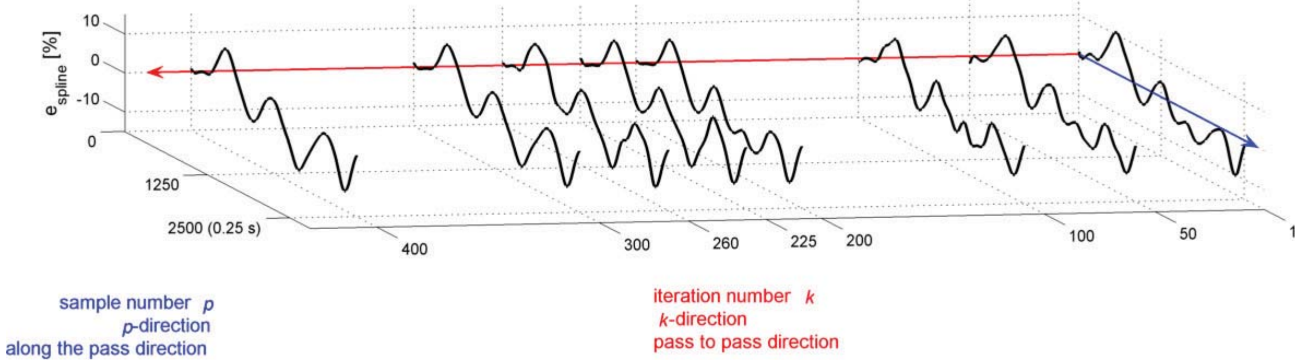
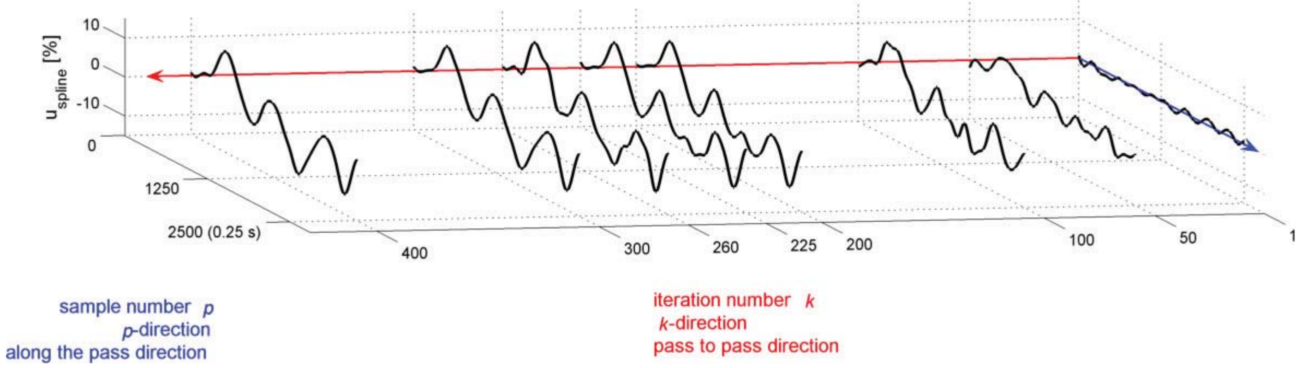


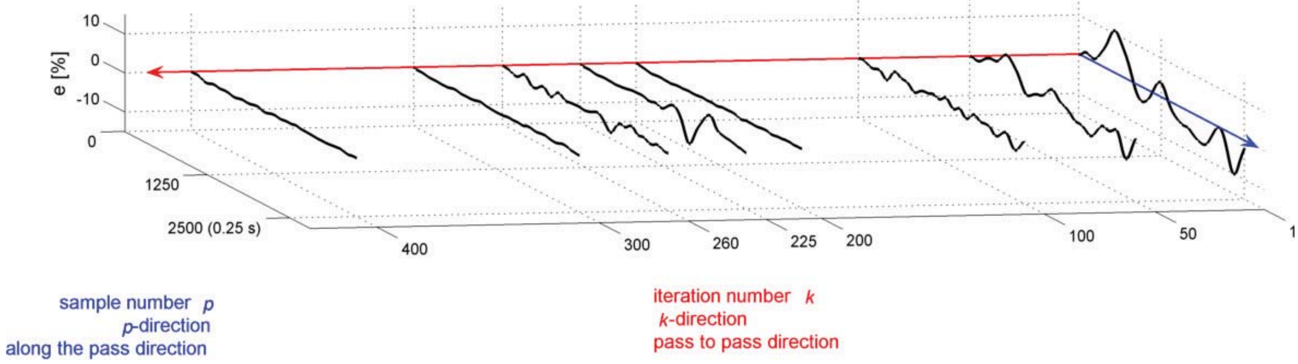
Fig. 6. Servo drive with PMSM and PSBRSC position controller input modification scheme



(a) Evolution of the PI controller input signal



(b) Evolution of the PSBRSC output signal



(c) Evolution of the process control error signal

Fig. 7. Evolution of the controller signals in position PI controller input modification scheme using the base value of $\alpha_{\max}^{\mathrm{ref}}$. The change of reference positions introduced at 201st pass.

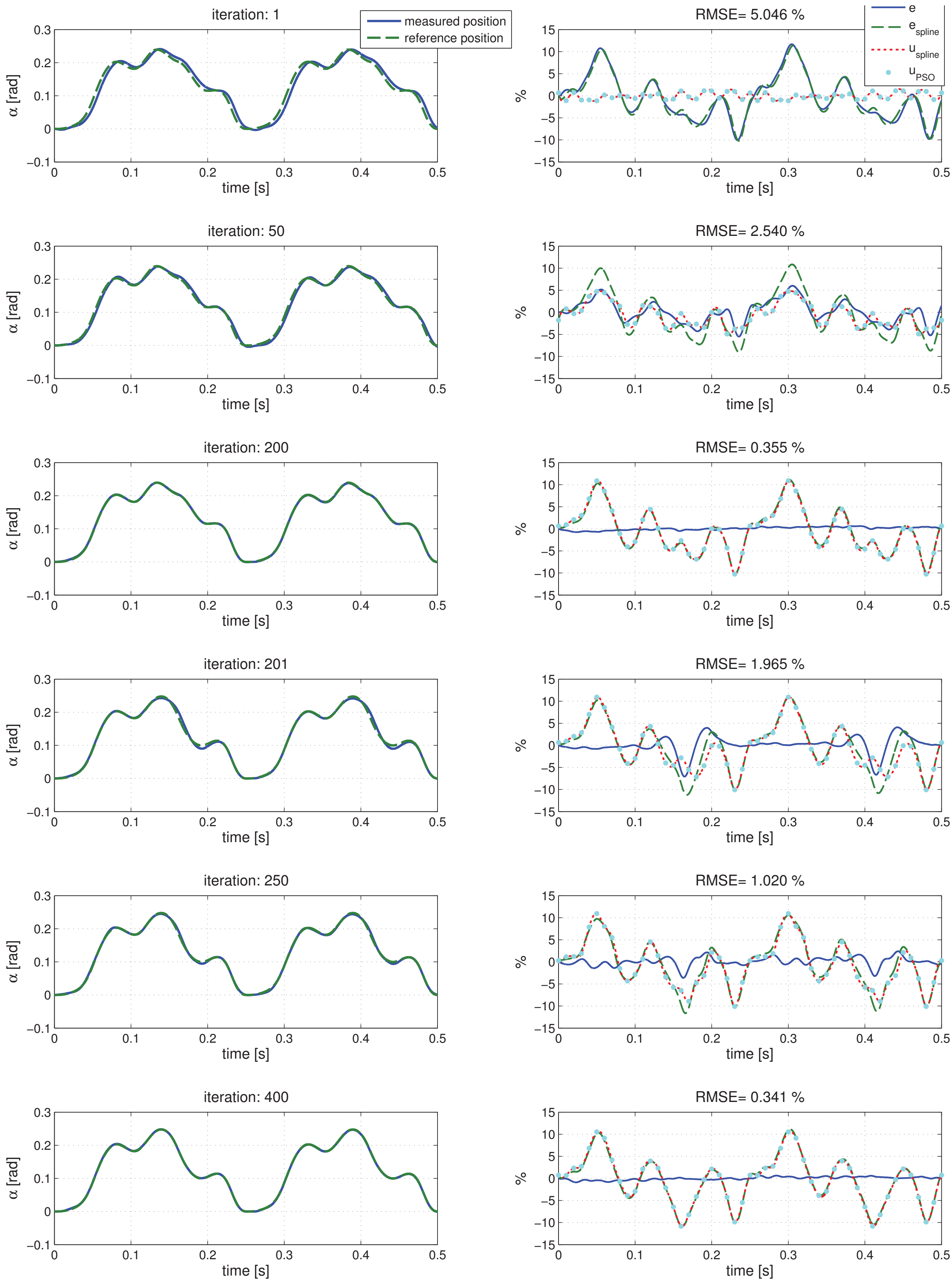


Fig. 8. Position PI controller input modification scheme. Reference and measured positions (left) and spline compensator output (u_{spline}), process control error (e) and PI controller input (e_{spline}) using the base value of $\alpha^{\mathrm{ref}}_{\mathrm{max}}$. The change of reference positions introduced at 201st pass.

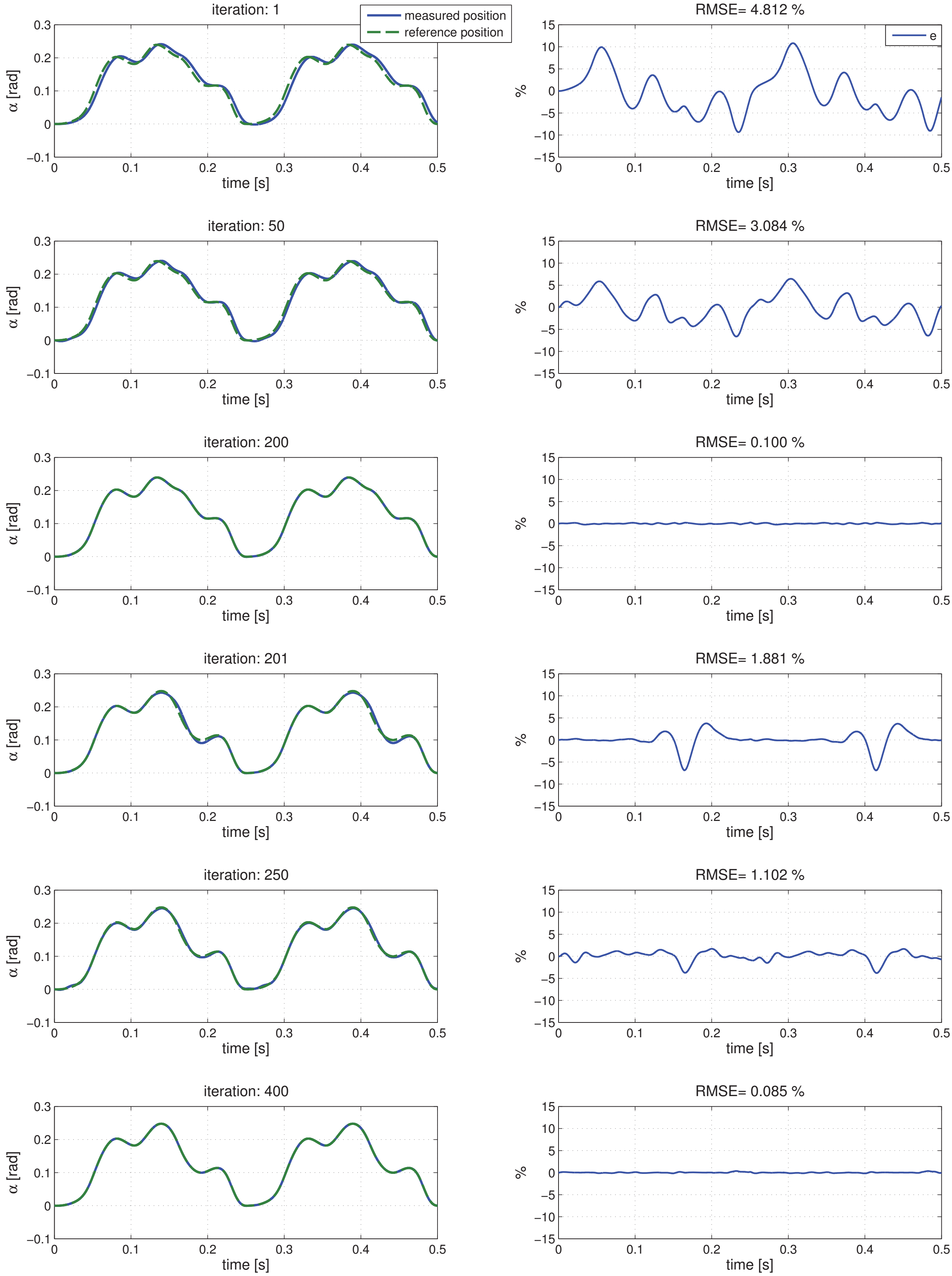


Fig. 9. Speed PI controller input modification scheme. Reference and measured positions (left) and process control error (e) using the base value of $\alpha_{\max}^{\mathrm{ref}}$. The change of reference positions introduced at 201st pass.

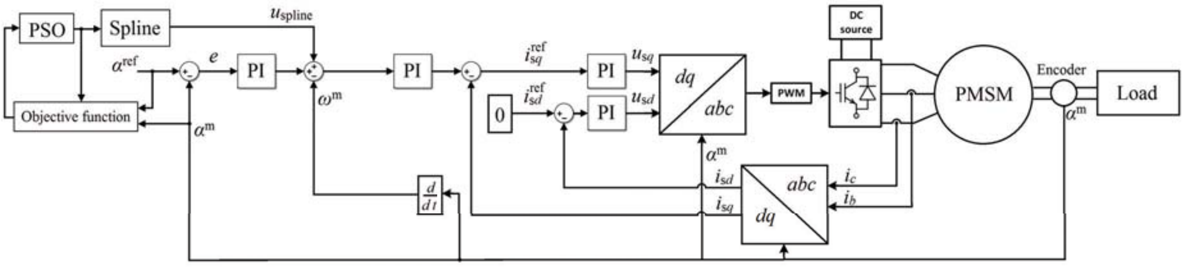


Fig. 10. Servo drive with PMSM and PSBRSC speed controller input modification scheme

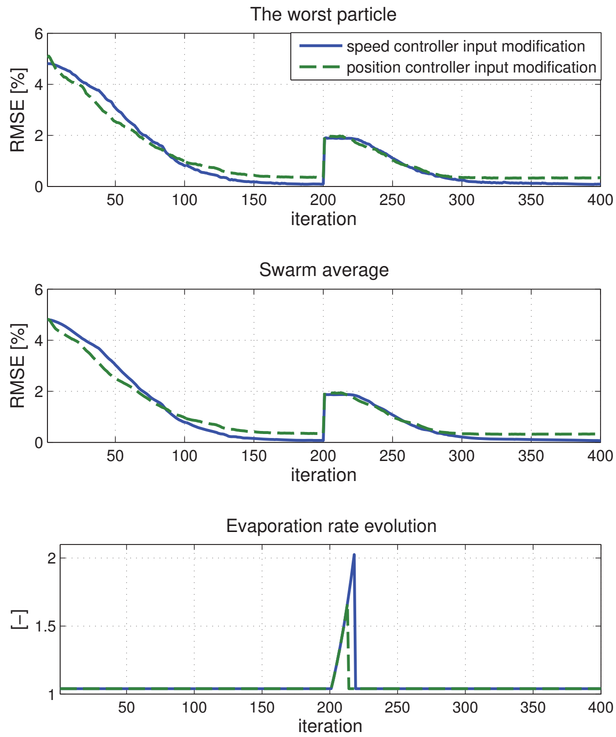


Fig. 11. RMSE evolution for the swarm average and the worst swarm in iteration using the base value of $\alpha^{\rm ref}_{\rm max}$ and evaporation rate evolution.

where $\mathbf{v}_j$ and $\mathbf{q}_j$ are the velocity and the position of the j-th particle, $\mathbf{q}_j^{\rm pbest}$ stores the best solution proposed so far by the j-th particle, $\mathbf{q}^{\rm gbest}$ denotes the best solution found so far, c_1, c_2, c_3 are the inertia, cognitive and social weights respectively. Velocity clamping is implemented and the maximum speed is $v_{\rm clmp}$. The random numbers $r^{\rm pbest}$ and $r^{\rm gbest}$ are uniformly distributed in the unit interval. In all the experiments described in this paper, the c_1, c_2 and c_3 factors were calculated using the constricted PSO formula [15, 14] and their values are 0.73, $0.73 \cdot 2.05$ and $0.73 \cdot 2.05$, respectively. The direction variable δ_p (-1 or 1) enables the swarms to switch between attract and repel modes and is chosen to be dimension-wise (p-wise), i.e. individual control of diversity is possible in each search dimension [17].

Evaporation rate growth mechanism

The evaporation rate growth mechanism has been implemented. If the sum of root mean square errors for all particles in the iteration *i* is more than two times higher than the sum of RMSE in the iteration *i*-1, which is supposed to be the result of an abrupt change in reference and/or disturbance, the evaporation rate begins to grow exponentially to provide a faster forgetting of the outdated best solution (9). After one of the particles settles in a new minimum (8), the evaporation constant returns to its base value in the next swarm iteration, which is shown in Fig. 11.

$$F(i+1) = \\ = \begin{cases} F(i)+1 & \text{if } \sum_{j=1}^{N_{\rm swarm}} \left(\mathrm{RMSE}\left(\mathbf{q}_j\left(i\right)\right)\right) > \\ & > 2\cdot\sum_{j=1}^{N_{\rm swarm}} \left(\mathrm{RMSE}\left(\mathbf{q}_j\left(i-1\right)\right)\right) \\ F(i)+1 & \text{if } F(i) > 0 \\ 0 & \text{if } \mathcal{J}\left(\mathbf{q}_j\left(i\right)\right) < \rho(i)P_j\left(i-1\right)\ , \end{cases} \tag{8}$$

$$\rho(i) = \rho_{\rm base}^{1+F(i)}, \tag{9}$$

where $\mathrm{RMSE}\left(\mathbf{q}_j\left(i\right)\right)$ is the root mean square of control errors along the pass of the j-th particle in i-th swarm iteration, $\mathcal{J}\left(\mathbf{q}_j\left(i\right)\right)$ is the current fitness of the j-th particle, $P_j = \mathcal{J}\left(\mathbf{q}_j^{\rm pbest}\right)$ stores the best personal fitness value so far of the j-th particle, $N_{\rm swarm}$ is the number of particles, $F(i)$ is a component of the exponent of evaporation rate in i-th swarm iteration, $\rho(i)$ denotes evaporation rate in i-th swarm iteration and ρ_{base} is the base value of evaporation rate.

Simulation model

A numerical model of the drive system with PMSM [1] has been built in Matlab/Simulink environment [18]. It has been assumed that the target application is the control of the base drive of an anthropomorphic robot in a process requiring a limited workspace. The control system is composed of a *d*-axis PI current controller (orthogonal *dq* coordinate system) and a PI cascade control system of current (*q*-axis), speed and position. The discrete PI controller with anti-windup shown in Fig. 4 has been adopted in the simulation. The PI current controllers have been tuned based on the optimal modulus criterion [1] and the PI speed controller has been tuned based on the symmetrical optimum criterion [1]. The gains of PI controller of position have been tuned by guessing and checking. The plant and PSO parameters are shown in Tab. 1.

Results

The presupposed scenario is as follows:

a) 400 swarm iterations,
b) changing the reference position, shown in Fig. 5, after 200 iterations,
c) swarms are initialized with a near zero $u_{\rm spline}$ (*k*=0) signal, which is shown in Fig. 2,
d) exogenous load of 60 % of nominal torque is applied between 0.15 seconds and 0.20 seconds time instances of the period,

e) viscosity contributes 25 % of nominal torque at nominal speed.

The selected waveforms of the e, e_{spline} and u_{spline} signals of the position controller modification scheme, which are presented in Fig. 7, illustrate the evolution of these signals. Fig. 8 shows two periods of e, e_{spline}, u_{spline} (samples coming directly from optimizer and remaining calculated using cubic spline interpolation) and the reference and measured positions at 1, 50, 200, 201, 250 and 400 swarm iterations for the position controller input modification scheme. Process control error converges to zero when the u_{spline} signal converges with the PI controller input (e_{spline}). Fig. 9 illustrates the evolution of the e signal and the reference and measured positions for the speed controller input modification scheme at the same iterations, as shown in Fig. 8. The evolution of the root mean square of the process control error samples for both schemes is shown in Fig. 11. The evaporation mechanism requires about 20 swarm iterations to forget an outdated best solution. When the evaporation rate growth mechanism was disabled, the algorithm needed from 50 to 100 iterations to effectively forget an obsolete best solution.

Comparison of alternative control schemes

Two alternative concepts of control scheme have been tested. In the first approach u_{spline} signal is added to position controller error, which is shown in Fig. 6. Speed controller input is modified by u_{spline} in the approach presented in Fig. 10. The obtained numerical simulation results suggest that the process control error in the speed controller input modification scheme can be reduced to a lower level as illustrated in Fig. 11. This might be caused by the less demanding shape of the objective function in a situation of modifying the cascade internal signal. Further investigation is required to confirm this theory.

Conclusions

A particle swarm based repetitive spline compensator for continuous repetitive processes has been developed. To maintain the globality of the optimization problem whilst reducing the dimensionality of the swarm, cubic spline interpolation of the PSO-based samples is introduced to produce controller input modification signal. The control task in the repetitive direction has been formulated in such a way that it poses a dynamic optimization problem. A novel evaporation rate growth mechanism, which improves the responsiveness of the abrupt change of the objective function, has been implemented. Two different compensation schemes have been tested and compared. It has been demonstrated that both proposed control methods decrease the root mean square of control errors along the pass.

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

REFERENCES

[1] Rumniak P., Ufnalski B., Grzesiak L. M.: Tuning of PI regulators in distributed control system for an electric vehicle, Przegląd Elektrotechniczny, vol. 9, pp. 290–294, 2015.
[2] Jingqing Han: From PID to Active Disturbance Rejection Control, IEEE Transactions on Industrial Electronics, 56(3), pp. 900–906, 2009.
[3] Namazi M.M., Borujeni M.M., Rashidi A., Nejad S.M.S., Jin-Woo Ahn: Torque ripple reduction of switched reluctance motor drive with adaptive sliding mode control and Particle Swarm Optimization, 2015 IEEE International Conference on Advanced Intelligent Mechatronics (AIM), 56(3), pp. 371-376, 7-11 July 2015.
[4] Dabkowski P., Gałkowski K., Rogers E., Cai Z., Freeman C. T., Lewin P. L.: Iterative Learning Control Based on Relaxed 2D Systems Stability Criteria, IEEE Transactions on Control Systems Technology, 21(3), pp. 1016-1023, 2013.
[5] Ufnalski B., Grzesiak L. M.: Repetitive neurocontroller with disturbance feedforward path active in the pass-to-pass direction for a VSI inverter with an output LC filter, Bulletin of the Polish Academy of Sciences Technical Sciences, vol. 64, iss. 1, pp. 115–125, March 2016.
[6] Ufnalski B., Grzesiak L. M.: Artificial neural network based voltage controller for the single phase true sine wave inverter – a repetitive control approach, Przegląd Elektrotechniczny, vol. 4, pp. 14–18, 2013.
[7] Wang Y., Gao F., Doyle F. J.: Survey on iterative learning control, repetitive control, and run-to-run control, Journal of Process Control, vol. 19, no. 10, pp. 1589-1600, 2009.
[8] Norrlof M.: An adaptive iterative learning control algorithm with experiments on an industrial robot, IEEE Transactions on Robotics and Automation, vol. 18, no. 2, pp. 245-251, 2002.
[9] Mandra S., Gałkowski K., Aschemann H., Rauh A.: Guaranteed cost iterative learning control - an application to control of permanent magnet synchronous motors, Proceedings of the 2015 IEEE 9th International Workshop on MultiDimensional (nD) Systems - nDS' 15, pp. 45-50, 2015.
[10] Kowalow D., Patan M., Paszke W., Romanek A.: Sequential design for model calibration in iterative learning control of DC motor, 20th International Conference on Methods and Models in Automation and Robotics (MMAR), 2015, pp. 794-799, Aug. 2015.
[11] Longman R. W.: Iterative/repetitive learning control: learning from theory, simulations, and experiments, Encyclopedia of the Sciences of Learning, pp. 1652-1657, 2012.
[12] ABB: ABB introduces new robotic laser cutting solutions, 2012, http://www.abb.pl/cawp/seitp202/7b167f23f577e56cc1257a41003ebd38.aspx
[13] Fine B. T.: Practical Iterative Learning Control: Intuitive Methods for Precision Motion Control, Ph.D. dissertation, University of California, Berkeley, 2009.
[14] Ufnalski B., Grzesiak L. M.: A plug-in direct particle swarm repetitive controller for a single-phase inverter, Przegląd Elektrotechniczny, vol. 6, pp. 6–11, 2014.
[15] Ufnalski B., Grzesiak L. M.: Plug-in direct particle swarm repetitive controller with a reduced dimensionality of a fitness landscape – a multi-swarm approach, Bulletin of the Polish Academy of Sciences Technical Sciences, vol. 63, iss. 4, pp. 857–866, Dec. 2015.
[16] Mathworks: Spline interpolation, 2016, http://www.mathworks.com/help/matlab/ref/spline.html
[17] Ufnalski B., Grzesiak L. M.: Plug-in Direct Multi-Swarm Repetitive Controller for the Sine Wave Inverter – on Keeping Particles Diversified in a Dynamic and Noisy Environment, 6th International Conference on Power Engineering, Energy and Electrical Drives 10th International Conference on Compatibility and Power Electronics (CPE-POWERENG), 2016, 29.06 – 01.07, (accepted to print).
[18] Ufnalski B., Basinski K.: Particle swarm based repetitive spline compensator, 2016, https://www.mathworks.com/matlabcentral/fileexchange/56448-particle-swarm-based-repetitive-spline-compensator

***Authors**: mgr inż. Karol Basiński, dr hab. inż. Bartłomiej Ufnalski, prof. dr hab. inż. Lech M. Grzesiak, Warsaw University of Technology, Institute of Control and Industrial Electronics, ul. Koszykowa 75, 00-662 Warszawa, Poland, email: {basinskk, bartlomiej.ufnalski, lech.grzesiak}@ee.pw.edu.pl*