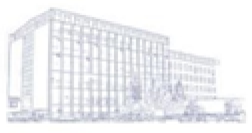
THE 5th INTERNATIONAL CONFERENCE
MECHANICAL ENGINEERING IN XXI CENTURY

December 9-10, 2020
Faculty of Mechanical Engineering in Niš, Serbia



A Survey of Swarm Intelligence-based Optimization Algorithms for Tuning of Cascade Control Systems: Concepts, Models and Applications

Zoran MILJKOVIĆ, Milica PETROVIĆ
University of Belgrade, Faculty of Mechanical Engineering, Department of Production Engineering, 11120 Belgrade
zmiljkovic@mas.bg.ac.rs, mmpetrovic@mas.bg.ac.rs

Abstract—**Nowadays, cascade control is still one of the most used control strategies in the manufacturing and process industries. The new requirements of precision and robustness of position and trajectory tracking in control systems for manufacturing components at micro-scale, influenced by hard nonlinearities such as friction and backlash, have motivated the effort toward the development of algorithms for optimal tuning of control parameters. This paper presents a literature review of the algorithms and methods used to solve this problem. Swarm intelligence inspired optimization algorithms, namely particle swarm optimization algorithm (PSO) and grey wolf optimization algorithm (GWO), are applied for tuning of P-PI cascade controllers of CNC machine tool servo system in the presence of friction and backlash. The objective of the optimization is to minimize the maximum position error during the reversal of the axes. A comparative analysis of proposed algorithms with a standard industry-based fine tune (FT) method is also provided. Simulation study as well as real-world experiments carried out on a CNC machine tool controller show a remarkable improvement in the performance of the cascade control system using the proposed swarm intelligence-based strategy.**

Keywords—**swarm intelligence, particle swarm optimization, grey wolf optimizer, cascade control systems**

I. INTRODUCTION

Cascade control systems are one of the most used control structures not only in process industries but also in manufacturing industries, particularly in the field of computer numerical control. Comparing to conventional single-loop feedback control systems, cascade control systems have the ability to correct or eliminate disturbances before they influence the controlled variable of interest and overcome the drawbacks when such disturbances related to the manipulated variable occur. A typical configuration of the cascade control system is based on two controllers within two nested loops, inner and outer loop. The controller in the inner loop is the primary (also called master) and the controller in the outer loop is secondary (aka slave). According to [1], the principal benefits of implementing such a control scheme are in the following two aspects (i) fast dynamic of the inner loop which can eliminate input disturbances, and (ii) improvement of the system response speed. On the other hand, their simple structure and easy way of implementation make them widely adopted in the industrial processes, especially in the process control industry and manufacturing industry.

Design and tuning of cascade control parameters are the most significant issues in this field. Therefore, many methods and approaches have been proposed to tune the parameters of controllers and achieve better control performance. One of the possible approaches that can be found in the state-of-the-art literature is based on the classical auto-tuning method. The authors in [2] proposed an on-line pattern recognition approach to acquire the parameters and expert system based on fuzzy logic inference to tune cascade control parameters of PID controllers online. The authors in [3] designed a cascade control system with the ability to tune one optimal parameter of PID controllers. Furthermore, an automatic tuning methodology was proposed in [4] in order to take into consideration high load disturbance rejection performance. Achieved simulation results lead towards conclusions that the proposed open–loop procedure allows advantages in terms of simultaneous tuning of the PID controllers as well as in the determination of a command signal.

On the other side, many approaches based on a single relay feedback test were reported in the relevant literature. The methodology for tuning parameters of the cascade control system was presented in [1], where two cases were analyzed (i) when process models are available and (ii) when process models are not available. According to the aforementioned methodology, the authors developed an auto-tuning method that uses relay feedback. As a result, phases of identification and tuning of the parameters are decoupled without the need for an extensive trial- and -error efforts for modeling and tuning. Furthermore, the authors in [5] proposed an auto-tuning procedure based on a single relay experiment. This procedure allows parameters for both loops to be identified simultaneously. The automatic tuning method based on a single relay experiment was proposed in [6]. The experimental results demonstrated the effectiveness of the tuning method when it is implemented in the domain of process control.

In other more contemporary works, algorithms are suggested for the direct design of cascade controllers driven by input-output data with a focus based on Virtual Feedback Tuning (VRFT). In this way, the experimental data serve to adjust the internal and external loops [7]. The automatic method that simultaneously tunes both PID controllers of the cascade control system was developed [8]. The method is based on a single closed-loop step test, while process information is represented by using the B-spline series. Optimization in the frequency domain with restrictions on maximum sensitivity, the limit of multiplicative uncertainty, and sensitivity to measurement noise is shown in [9]. However, these works are applied to systems with slow dynamics and with requirements of precision and quality in the dynamic response that are not very demanding. A large variety of control techniques such as predictive control and sliding control have been reported in the literature [10], [11], and fuzzy and neuro-fuzzy control techniques have been reported in [12], [13].

According to the presented literature review, optimal tuning of cascade control systems' parameters becomes a cumbersome task in the presence of hard nonlinearities (friction, backslash, stiction). In the last two decades, nature-inspired swarm intelligence metaheuristic algorithms have been used as very efficient techniques for obtaining the optimal solutions of high-dimensional, nonlinear, and complex optimization problems [14]. The popularity of those algorithms is due to several main reasons [15]: simplicity in concept, flexibility to adapt to different problems, gradient-free mechanism, and ability to avoid local optima. Therefore, some of the metaheuristic methods were applied in the tuning of the controllers for cascade control systems. A multi-objective GA for optimal tuning of a networked linear controller applied to control a high-performance drilling process was proposed in [16]. The tool's working life and the material removal rate are criteria to be maximized. Furthermore, investigations in this field have been continued and authors in [13] implemented a hybrid approach based on an adaptive neuro-fuzzy inference system for modeling and control of the cutting force during the high-performance drilling process. An evolutionary algorithm called differential evolution (DE) was applied in [17] to tune the parameters of the adaptive cascade controller. Although the DE algorithm demonstrated effectiveness in trajectory tracking control of the hydraulic actuator, practical implementation of the proposed controller was not presented in the paper. In literature [18], a novel swarm intelligence-based Whale Optimization Algorithm (WOA) is applied for optimal design and tuning parameters of fuzzy control systems.

The particle swarm optimization (PSO) algorithm is a population-based method originally proposed in [19]. It has been proven to be a powerful tool for solving global engineering optimization problems. Compared to other algorithms, the advantages of the PSO algorithm lie in its easy programming and implementation, fast convergence speed, and effective performance [20], [21], [22]. These advantages motivated us to introduce the PSO algorithm to optimize the parameters of servosystem influenced by hard nonlinearities such as friction and backlash. Another motivation comes from a literature survey of current state-of-the-art methods. Although the authors in [23] presented a cascade control system where the inner and outer loop of the controllers are tuned by the PSO algorithm, to the best of authors' knowledge, none of the literature sources in the field has carried out the influence of the hard nonlinearities (friction and backslash) on cascade controllers.

On the other side, a grey wolf optimizer (GWO) is a novel population-based method proposed in [15]. According to [24], the main advantages of this method are (i) simple and free from computational burden, (ii) flexible to apply for different problems without any special modification of its structure (iii) better capability to avoid local optima in comparison to the conventional optimization algorithm, (iv) easy to transform into the programming language and implement.

Therefore, in this paper, PSO and GWO algorithms are applied to identify optimal parameters of servosystem influenced by hard nonlinearities such as friction and backlash, as well as to provide near-optimal solutions of P/PI cascade controllers' parameters. The main objective is to minimize the maximum position error while maintaining accuracy and without significantly increasing the control effort. The performance of the proposed PSO and GWO algorithms are experimentally verified and compared with the standard industry approach called Fine Tune (FT).

The structure of the paper is organized as follows. The literature overview of the cascade controllers and approaches used to model and optimize those systems are presented in introductory Section 1. Section 2 introduces a problem formulation with the mechanical modeling of the proposed system. Section 3 gives a description of the PSO algorithm, while Section 4 presents the GWO algorithm applied for tuning of P/PI controller parameters. The experimental results are given in Section 5. The concluding remarks of the paper are summarized in Section 6.

II. Problem formulation

The drive system of the CNC machine tool analyzed in this paper is based on two masses (motor and load) linked with the both elastic and dumping connection. The motor coupled to load (Fig. 1) is represented as a two-mass oscillator, where the first mass represents the moment of inertia of the drive and the second mass refers to the moment of inertia of the load side. The elastic and dumping components are modeled by spring and dumper, respectively. The parameters of this mechanical model are as follows:

- K is the torsion spring constant;
- B is the inner damping coefficient of the shaft;
- J_M is the motor moment of inertia;
- J_L is the load moment of inertia;
- M_M is the motor torque;
- M_L is the load torque disturbance;
- M_S is the transmitted shaft torque.

In the model used in this paper, the motor angular velocity (ω_M), and load angular velocity (ω_L), as well as the transmitted shaft torque (M_S), are used as state variables.

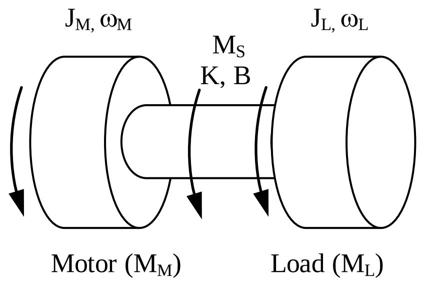


Fig. 1. Model of the two-mass system

After a series of transformations, the transfer function of motor angular velocity (ω_M) to the motor torque (M_M) is given by equation (1):

$$H_{\omega_M/M_M}(s) = \frac{1}{J_M s} \cdot \frac{s^2 + 2D_1\omega_{01}s + \omega_{01}^2}{s^2 + 2D_2\omega_{02}s + \omega_{02}^2} \qquad (1)$$

where characteristic angular frequencies ω_{01} and ω_{02} are introduced as (2) and (3):

$$\omega_{01} = \sqrt{K / J_L}, \qquad (2)$$

$$\omega_{02} = \omega_{01}\sqrt{1 + (J_L / J_M)}, \qquad (3)$$

and damping coefficients D_1 and D_2 are given by (4) and (5):

$$D_1 = \frac{B}{2\omega_{01}J_L}, \qquad (4)$$

$$D_2 = D_1\sqrt{1 + \frac{J_L}{J_M}} = \frac{B(J_M + J_L)}{2J_M J_L \omega_{02}}. \qquad (5)$$

Furthermore, the transfer function between angular velocity of the load (ω_L) and angular velocity of the motor (ω_M) is presented by equation (6):

$$H_{\omega_L/\omega_M}(s) = \frac{2D_1\omega_{01}s + \omega_{01}^2}{s^2 + 2D_1\omega_{01}s + \omega_{01}^2} \qquad (6)$$

A. Friction

Friction is a phenomenon that has a strong influence on the performance and behavior of both mechanical and electromechanical systems. Since the effect of friction can lead to significant error of CNC machine tools positioning system, good representation and prediction of friction force are important for control of such kind of electromechanical systems. Therefore, in order to compensate the error caused by the friction force, the combined Coulomb-viscous model of friction force F is adopted (equation 7):

$$F = F_C \,\mathrm{sgn}(v) + F_V v \qquad (7)$$

where F is the friction force, v is the relative velocity between two surfaces in contact, F_C is the Coulomb friction level, and component F_V represents a small viscous friction.

B. Backlash

Besides friction, the backlash is another common nonlinear factor that affects the behavior of the CNC machine tools positioning systems. In order to model this nonlinearity, the classic dead zone model is utilized in this paper. The exponential model for leads-crew backlash compensation is performed according to the following equation (8):

$$R_P = PP_2\, e^{-t/PP_3} \qquad (8)$$

where PP_2 and PP_3 are backlash peak amplitude and backlash peak time period, respectively.

In order to model the phenomenon of hysteresis nonlinearity in the actuator, parameter f_H which represents the amplitude of the hysteresis is adopted in the proposed control system.

C. Fitness function

According to the aforementioned analysis, a target function for optimization is represented by the following six parameters, equation (9):

$$K = \left[K_p^{pos}\ K_p^{vel}\ K_i^{vel}\ PP_2\ PP_3\ f_H\right] \qquad (9)$$

where K_p^{pos} represents the proportional gain of the position controller in the outer loop, K_i^{vel} and K_p^{vel} represent the proportional and integral gain of the speed controller in the inner loop, PP_2 and PP_3 are the backlash peak amplitude and the backlash peak time compensators, respectively, and f_H is the parameter that compensates the friction hysteresis.

The main objective of this research is to minimize the maximum position error E_{pk} (equation 10) which is caused by changing the direction of the axis or by changing the trajectory of the movement. E_{pk} is directly influenced by hard nonlinearities of the mechanical systems such as friction and backlash and its minimization is of high importance for the improvement of the product quality.

$$K = \left[K_p^{pos}\ K_p^{vel}\ K_i^{vel}\ PP_2\ PP_3\ f_H\right]_{OPT} = \arg\min\left(\max(E_{pk})\right) \qquad (10)$$

III. Particle swarm optimization algorithm

Particle swarm optimization (PSO) algorithm, originally proposed in [19], is a population-based optimization method inspired by the movement and intelligence of the organisms in a swarm (e.g., a flock of birds, or school of fish). In order to search for food, each member of the swarm determines its velocity based on their personal experience as well as information gained through interaction with other members of the swarm.

Traditional PSO algorithm is initialized with a population of randomly generated candidate solutions known as particles. Each particle flies through the multidimensional search space of the optimization problem with a specific velocity searching for the optimal solution; its position represents a potential solution of the problem and its velocity is dynamically adjusted according to its own flying experience and according to the neighbouring flying experience. Particle position and particle velocity are updated iteratively by using equation (11) and equation (12):

$$V_{id}^{t+1} = W \cdot V_{id}^{t} + C_1 \cdot rand() \cdot (P_{ld}^{t} - X_{id}^{t}) + C_2 \cdot Rand() \cdot (P_{gd}^{t} - X_{id}^{t}) \qquad (11)$$

$$X_{id}^{t+1} = X_{id}^{t} + V_{id}^{t+1} \qquad (12)$$

$$W = W_{max} - \frac{W_{max} - W_{min}}{iter_{max}} \cdot iter \qquad (13)$$

where:

- t is the generation number;
- represents the velocity of the particle i in generation t,
- represents the velocity of the particle i in generation t+1;
- represents the position of the particle i in generation t;
- represent the positions of the particle i in generation t+1;
- is the local best solution ("pbest") of each particle;
- is the global best solution ("gbest") of the swarm;
- W is inertia weight set as in equation (13);
- C_1 and C_2 are positive acceleration constants;
- rand() and Rand() are two random numbers in the range [0,1].

The pseudocode of PSO algorithm implemented to optimize fitness function given by equation (10) is described in Table I [25]:

TABLE I. PSEUDO CODE OF PSO ALGORITHM

```
Initialize the parameters of PSO algorithm (swarm size, maximum number of generation, inertia weights, $W_{max}$ and $W_{min}$, acceleration constants, $C_1$ and $C_2$);
Initialize a swarm of particles with random positions and velocities (equations 14 and 15);
Evaluate each particle's fitness function by using (equation 10);
Initialize the global ("gbest") and the local best position ("pbest");
Repeat
    generation = generation + 1;
    generate next swarm by updating the velocities and positions of the particles;
    evaluate swarm;
    compute each particle's fitness function (equation 10);
    find new global ("gbest") and the local best position ("pbest");
    update "gbest" of the swarm and "pbest" of each particle;
Until the maximum of generation is not met
Output: the optimal parameters
$\left[K_p^{pos}\ K_p^{vel}\ K_i^{vel}\ PP_2\ PP_3\ f_H\right]_{OPT}$
```

IV. Grey wolf optimization algorithm

Grey Wolf Optimization (GWO) algorithm, initially proposed in [15], belongs to a class of novel swarm-based meta-heuristics inspired by the social leadership and hunting technique of grey wolves in nature. According to the GWO algorithm, the grey wolves are classified into four levels of social hierarchy: alpha (α), beta (β), delta (δ), and omega (ω). The alphas are the leaders of the group responsible for the hunting process and making decisions. The betas belong to the second level of the hierarchy and they assist the alphas in making decisions, while deltas belong to the third level and dominate the wolves of the last level omega. The omegas are the lowest ranking grey wolves on the pyramid of social hierarchy.

GWO algorithm is based on the aforementioned social behavior of the grey wolves. The optimization (hunting) process is initialized with randomly generated candidate solutions (grey wolves) in a multi-dimensional search space. This phase of searching for prey is also known as exploration. The best fitness solution is defined as alpha (α), the second and third best solutions are beta (β) and delta (δ), and the rest of the solutions are assumed to be omega (ω). In order to catch the prey, the α, β, and δ grey wolves firstly encircle the victim. During the optimization (hunting) process, they estimate the victim position and update their positions randomly around the victim according to the mathematical model given by the following equations (14) and (15):

$$D = \left|\vec{C} \cdot \vec{X}_p(\mathrm{t}) - \vec{X}(\mathrm{t})\right| \tag{14}$$

$$\vec{X}(\mathrm{t+1}) = \vec{X}_p(\mathrm{t}) - \vec{\mathrm{A}} \cdot \vec{\mathrm{D}} \tag{15}$$

where t represents iteration, $\vec{X}_p$ is the position vector of the prey, $\vec{X}$ is the position vector of a grey wolf, $\vec{A}$ and $\vec{C}$ are coefficient vectors calculated by (16), (17):

$$\vec{A} = 2\vec{a} \cdot \vec{r}_1 - \vec{a} \tag{16}$$

$$\vec{C} = 2 \cdot \vec{r}_2 \tag{17}$$

The components of vector $\vec{a}$ linearly decrease from 2 to 0, and r_1 and r_2 are random vectors in [0,1].

Furthermore, hunting behavior of the grey wolves can be mathematically modeled by equations (18), (19) and (20):

$$D_\alpha = \left|\vec{C}_1 \cdot \vec{X}_\alpha - \vec{X}\right| \tag{18}$$

$$D_\beta = \left|\vec{C}_1 \cdot \vec{X}_\beta - \vec{X}\right| \tag{19}$$

$$D_\delta = \left|\vec{C}_1 \cdot \vec{X}_\delta - \vec{X}\right| \tag{20}$$

The positions of α, β, and δ grey wolves (the first free best solutions) are updated according to the following formulas (21), (22), (23) and (24):

$$\vec{X}_1 = \vec{X}_\alpha - \vec{A}_1(\vec{\mathrm{D}}_\alpha) \tag{21}$$

$$\vec{X}_2 = \vec{X}_\beta - \vec{A}_2(\vec{\mathrm{D}}_\beta) \tag{22}$$

$$\vec{X}_3 = \vec{X}_\delta - \vec{A}_3(\vec{\mathrm{D}}_\delta) \tag{23}$$

$$\vec{X}(\mathrm{t+1}) = \frac{\vec{X}_1 + \vec{X}_2 + \vec{X}_3}{3} \tag{24}$$

Finally, the hunting process is finished by attaching the prey (exploitation phase). The pseudocode of the GWO algorithm is presented in Table II:

TABLE II. PSEUDO CODE OF GWO ALGORITHM

```
Initialize the parameters of GWO algorithm (population size, maximum number of iterations, position vector X, and vectors A, a, C);
Initialize a population of grey wolves (equations 14 and 15);
Evaluate each grey wolf's fitness function by using (equation 10);
Identify three best wolves (the best search agent - $X_\alpha$, the second best search agent - $X_\beta$, the third best search agent - $X_\delta$) according to their fitness functions;
Repeat
    generate next population by updating each agent position (21-23);
    update a, A, C by using (16) and (17);
    compute each agent's fitness function (equation 10);
    update $X_\alpha$, $X_\beta$, $X_\delta$;
Until the maximum of generation is not met
Output: the optimal parameters
    $\left[K_p^{pos}\ K_p^{vel}\ K_i^{vel}\ PP_2\ PP_3\ f_H\right]_{OPT}$
```

V. EXPERIMENTAL RESULTS

In order to validate the proposed optimization methodology, experimental simulations are performed in Matlab software package. The parameters of the PSO algorithm are set as follows: the size of population is 20, the maximum number of generations is 100, the inertia weight W is set starting with 1.1 and is linearly decreased to 0.1. Acceleration constants C_1 and C_2 are set to 2.0. The parameters for the GWO algorithm are set as follows: the size of population is 20, the maximum number of iterations is 100, the parameter *a* linearly decreased from 2 to 0, and r_1 and r_2 are random vectors in [0,1]. The simulation results of the PSO and GWO algorithm are compared with the results achieved by one of the standard industry approaches used to manually tune CNC machine tools - Fine Tune (FT) method. After the performing optimization process, the achieved optimal values of the six parameters, $\left[K_p^{pos}\ K_p^{vel}\ K_i^{vel}\ PP_2\ PP_3\ f_H\right]$, are presented in Table III.

TABLE III. CONTROL PARAMETERS

Control parameters	**Optimization method**		
	Fine Tune method	***PSO algorithm***	***Grey Wolf Optimizer***
K_p^{pos}	66.6667	74.5	75
K_p^{vel}	0.2865	0.4983	0.2632
K_i^{vel}	0.0080	0.002401	0.0012
PP_2	0.7184	0.0902	0.4368
PP_3	0.0080	7.6635e-05	0.04393
f_H	0.1288	0.00596	0.00231

Fig. 2 shows experimental comparisons of the FT method, PSO, and GWO approach for three merit functions, maximum position error (maxE), the accuracy (ITAE), and the control effort (IAU), respectively. As might be seen from Fig. 2, the GWO algorithm achieves a significant improvement in minimization of maximum error (66.4%) comparing with the FT method. The second-best result is achieved by the PSO algorithm with an improvement of (62.5%) comparing to the FT method. However, even though the position error achieved using swarm intelligence-based optimization algorithms is less, results reported in Fig. 2 show that the accuracy (ITAE) of the FT method was better in relation to the other two methods with less the control effort (IAU) required to follow the desired trajectory.

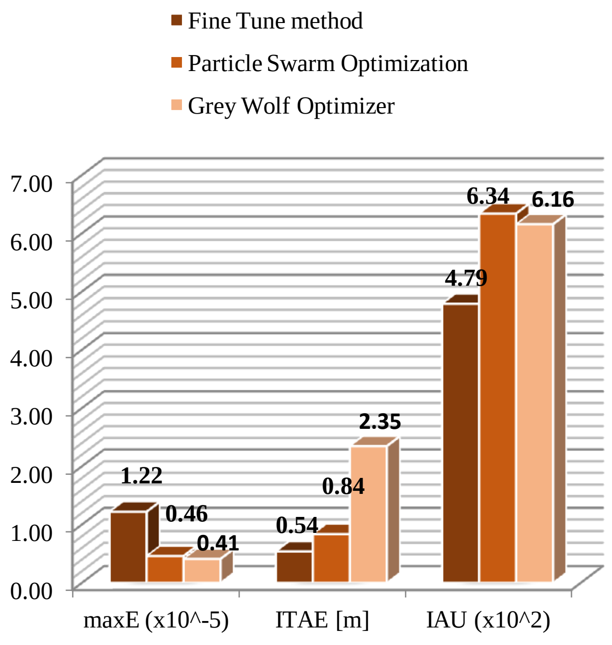


Fig. 2. Simulation results

VI. CONCLUSIONS

This paper presents two biologically inspired swarm intelligence-based optimization algorithms, namely particle swarm optimization algorithm (PSO) and grey wolf optimization (GWO) algorithm, implemented to optimally adjust parameters of the P-PI cascade controllers for CNC machine tool positioning system. Both the algorithms are proposed in order to simultaneously tune a proportional position controller in the external loop and a proportional-integral speed controller in the internal loop of the proposed cascade control system in the presence of nonlinearities. The minimization of the maximum position error directly influenced by nonlinearities such as friction and backlash is the main optimization objective of this work. The proposed optimization procedure results in a set of six optimal parameters of the servosystem: the proportional gain of the outer loop, the proportional and integral gain of the inner loop, the backlash peak amplitude, the backlash peak time, and the friction hysteresis parameter. The proposed PSO and GWO algorithms are implemented in Matlab software environment and experimental results are compared with ones obtained by applying the industry-driven methods named Fine Tune (FT) method. The achieved experimental results show a very important improvement in the tuning and behavior of the control system by minimization of the peak of the trajectory error. The advantage of the GWO over the FT method through the improvement of the maximum peak error is 66.4%, while the improvement of the PSO algorithm is 62.5%. Although the achieved position error (maxE) using swarm intelligence based optimization algorithms is less, better results in terms of accuracy (ITAE) are obtained using the

Fine Tune method. One of the future research directions could be oriented towards the implementation of multi-objective swarm intelligence-based optimization algorithms for the optimal design of cascade control systems.

Acknowledgment

This work has been financially supported by the Ministry of Education, Science and Technological Development of the Serbian Government, through the project "Integrated research in macro, micro, and nano mechanical engineering – Deep learning of intelligent manufacturing systems in production engineering", under the contract number 451-03-68/2020-14/200105, and by the Science Fund of the Republic of Serbia, Grant No. 6523109, AI - MISSION4.0, 2020-2022.

References

[1] H. P. Huang, I. L. Chien, Y. C. Lee, and G. Bin Wang, "A simple method for tuning cascade control systems," *Chem. Eng. Commun.*, vol. 165, no. 1, pp. 89–121, 1998, doi: 10.1080/00986449808912371.

[2] M. X. Li, P. M. Bruijn, and H. B. Verbruggen, "Tuning cascade PID controllers using fuzzy logic," *Math. Comput. Simul.*, vol. 37, no. 2–3, pp. 143–151, 1994, doi: 10.1016/0378-4754(94)00003-4.

[3] F. S. Wang, W. S. Juang, and C. T. Chan, "Optimal tuning of PID controllers for single and cascade control loops," *Chem. Eng. Commun.*, vol. 132, no. 1, pp. 15–34, 1995.

[4] A. Visioli and A. Piazzi, "An automatic tuning method for cascade control systems," *Proc. IEEE Int. Conf. Control Appl.*, vol. 2, pp. 2968–2973, 2006, doi: 10.1109/CACSD-CCA-ISIC.2006.4777110.

[5] S. Song, W. Cai, and Y. G. Wang, "Auto-tuning of cascade control systems," *ISA Trans.*, vol. 42, no. 1, pp. 63–72, 2003, doi: 10.1016/s0019-0578(07)60114-1.

[6] A. Leva and F. Donida, "Autotuning in cascaded systems based on a single relay experiment," *J. Process Control*, vol. 19, no. 5, pp. 896–905, 2009, doi: 10.1016/j.jprocont.2008.11.013.

[7] S. Formentin, A. Cologni, D. Belloli, F. Previdi, and S. M. Savaresi, *Fast tuning of cascade control systems*, vol. 44, no. 1 PART 1. IFAC, 2011.

[8] J. C. Jeng and M. W. Lee, "Simultaneous automatic tuning of cascade control systems from closed-loop step response data," *J. Process Control*, vol. 22, no. 6, pp. 1020–1033, 2012, doi: 10.1016/j.jprocont.2012.04.010.

[9] A. I. Ribić and M. R. Matausek, "An analysis, design and tuning of Cascade Control Systems in the presence of constraints in actuator and process outputs," *J. Process Control*, vol. 24, no. 12, pp. 7–17, 2014, doi: 10.1016/j.jprocont.2014.09.014.

[10] P. J. Serkies and K. Szabat, "Application of the MPC to the position control of the two-mass drive system," *IEEE Trans. Ind. Electron.*, vol. 60, no. 9, pp. 3679–3688, 2013, doi: 10.1109/TIE.2012.2208435.

[11] K. Szabat, T. Orowska-Kowalska, and P. Serkies, "Robust Control of the Two-mass Drive System Using Model Predictive Control," *Robust Control. Theory Appl.*, 2011, doi: 10.5772/15730.

[12] R. E. Haber, J. R. Alique, A. Alique, and R. H. Haber, "Controlling a complex electromechanical process on the basis of a neurofuzzy approach," *Futur. Gener. Comput. Syst.*, vol. 21, no. 7, pp. 1083–1095, 2005, doi: 10.1016/j.future.2004.03.008.

[13] A. G. Martin and R. E. H. Guerra, "Internal model control based on a neurofuzzy system for network applications. a case study on the high-performance drilling process," *IEEE Trans. Autom. Sci. Eng.*, vol. 6, no. 2, pp. 367–372, 2009, doi: 10.1109/TASE.2008.2006686.

[14] M. Petrović and Z. Miljković, "Grey Wolf Optimization Algorithm for Single Mobile Robot Scheduling," *Proc. 4th Int. Conf. Electr. Electron. Comput. Eng. IcETRAN*, p. ROI1.2.1-6, 2017.

[15] S. Mirjalili, S. M. Mirjalili, and A. Lewis, "Grey Wolf Optimizer," *Adv. Eng. Softw.*, vol. 69, pp. 46–61, 2014, doi: 10.1016/j.advengsoft.2013.12.007.

[16] D. Martin, R. del Toro, R. Haber, and J. Dorronsoro, "Optimal tuning of a networked linear controller using a multi-objective genetic algorithm and its application to one complex electromechanical process," *Int. J. Innov. Comput. Inf. Control*, vol. 5, no. 10(B), pp. 3405–3414, 2009.

[17] L. D. S. Coelho and M. A. B. Cunha, "Adaptive cascade control of a hydraulic actuator with an adaptive dead-zone compensation and optimization based on evolutionary algorithms," *Expert Syst. Appl.*, vol. 38, no. 10, pp. 12262–12269, 2011, doi: 10.1016/j.eswa.2011.04.004.

[18] R. C. David, R. E. Precup, S. Preitl, E. M. Petriu, A. I. Szedlak-Stinean, and R. C. Roman, "Design of low-cost fuzzy controllers with reduced parametric sensitivity based on whale optimization algorithm," *28th Mediterr. Conf. Control Autom. (MED). IEEE*, pp. 440–445, 2020, doi: 10.1109/FUZZ48607.2020.9177536.

[19] J. Kennedy and R. Eberhart, "Particle swarm optimization," *Proc. IEEE Int. Conf. Neural Netw.*, pp. 1942–1948, 1995, doi: 10.1007/978-3-642-37846-1_3.

[20] M. Petrović, N. Vuković, M. Mitić, and Z. Miljković, "Integration of process planning and scheduling using chaotic particle swarm optimization algorithm," *Expert Syst. Appl.*, vol. 64, 2016, doi: 10.1016/j.eswa.2016.08.019.

[21] M. Petrović, M. Mitić, N. Vuković, and Z. Miljković, "Chaotic particle swarm optimization algorithm for flexible process planning," *Int. J. Adv. Manuf. Technol.*, vol. 85, no. 9–12, 2016, doi: 10.1007/s00170-015-7991-4.

[22] Z. Miljković and M. Petrović, "Application of modified multi-objective particle swarm optimisation algorithm for flexible process planning problem," *Int. J. Comput. Integr. Manuf.*, 2016, doi: 10.1080/0951192X.2016.1145804.

[23] A. L. Sangeetha, N. Bharathi, A. B. Ganesh, and T. K. Radhakrishnan, "Particle swarm optimization tuned cascade control system in an Internet of Things (IoT) environment," *Measurement*, vol. 117, pp. 80–89, 2018, doi: 10.1016/j.measurement.2017.12.014.

[24] B. Nayak, A. Mohapatra, and K. B. Mohanty, "Parameter estimation of single diode PV module based on GWO algorithm," *Renew. Energy Focus*, vol. 30, no. 00, pp. 1–12, 2019, doi: 10.1016/j.ref.2019.04.003.

[25] M. Petrovic, A. Villalonga, Z. Miljkovic, F. Castano, S. Strzelczak, and R. Haber, "Optimal tuning of cascade controllers for feed drive systems using particle swarm optimization," *IEEE Int. Conf. Ind. Informatics*, pp. 325–330, 2019, doi: 10.1109/INDIN41052.2019.8972132.