

Elastic FOPID+FIR Controller Design Using Hybrid Population-Based Algorithm

Krystian Łapa

Abstract In this paper a new method for elastic H_∞ -optimal fractional order PID with FIR filters (FOPID + FIR) controller design using hybrid population-based algorithm is presented. With the use of a population-based algorithm an initial structure of the controller is adjusted in a such way that the designed controller fulfills the control objective in the best way possible. Moreover, in the control process the controller feedback signals' noise and discretization were taken into consideration. The goal of this paper is to show the influence of using FIR filters and FOPID controller structure on accuracy and to present possibilities of designing elastic controller structure using proposed hybrid population-based algorithm. The proposed method was tested on typical control problem.

Keywords PID controller FOPID controller • FIR filter Hybrid algorithms

1 Introduction

The problem of designing control systems is well known in the literature [1]. This is due to the fact that the quality of work of individual parts or even of entire machines mainly depends on the characteristics of the used controller. The proper controller design should take following elements under consideration: indication of measurable signals, selection of the controller structure, tuning of controller parameters and implementation in target hardware platform with fulfillment of requirements of real-time work. Usually these steps are performed in the presented order.

In the literature there are well-known controller structures such as: controllers structures based on the combination of linear correction terms, e.g. PID controllers (optionally with gain scheduling algorithm, with feed-forward path or with additional low-pass filters [2]), state feedback controllers, nonlinear controllers based on

K. Łapa (✉)

Institute of Computational Intelligence, Częstochowa University of Technology,
Częstochowa, Poland
e-mail: krystian.lapa@iisi.pcz.pl

computational intelligence and hybrid controllers, in which are combined approaches from other groups. However, in practice PID controllers are used the most often [1]. It is a result of widespread knowledge of how they work and their relatively simple implementation in a microprocessor-based control systems. An extension of PID controllers are, among the others, H^∞ -optimal (in these methods the problem of control is defined as the optimal control task, and then the controller, which may perform such a task, is designed) Fractional Order PID (FOPID) controllers. These controllers improve responses with respect to rational-type controllers such as PID [4].

The problems associated with designing of the controller apply, among the others, to: difficult and time consuming design process, modification of the structure to support more than one feedback signal (typical PID controllers consist of single PID block for processing a single controller signal), proper selection of the structure parameters, noise reduction, not taking into considering discretization of the feedback signals, etc.

Among the experimental methods for the design of control systems, methods based on artificial intelligence [3] and in particular the methods of evolution [5] are becoming more common. The methods of evolution are based on populations of the solutions, where each solution can represent the structure and the parameters of the single controller [8]. During evolution the population can improve (better solutions are being find) by modifying or mixing system structure and parameters between solutions. This process is usually based on fitness function value calculated for each solution.

In this paper a new method for elastic H^∞ -optimal fractional order PID with FIR filters (FOPID + FIR) controller design using hybrid population-based algorithm is presented. Low-pass finite-impulse-response filters (FIR) with programmable characteristics for each of the measurement signals are used in the feedback loop. These filters are designed to suppress interference that could disrupt the work of the control system. Due to that it is possible to find the structure and parameters of the controller, which makes it immune to this type of interference. From the other hand, using proposed elastic FOPID controller instead of standard PID controller allow to handle higher order processes by performing optimization with various integral performance indices. In the design process an universal initial structure is proposed, which in the process of evolution will be adjusted in a way that the designed controller fulfills the control objective in the best way possible. This elastic structure consists of FOPID functional blocks and FIR filters, both with programmable structures, connections and parameters. Due to this approach the design of the control system can be regarded as one continuous process, unlike the commonly used method of trial and error. As a result, the process of controller design is performed easier and faster. Details of the proposed method are described in Sect. 3.

This paper is organized into 5 sections. Section 2 contains a description of the elastic FOPID + FIR controller structure, while Sect. 3 shows the proposed evolutionary algorithm used to design control system. Simulation results are presented in Sect. 4. Conclusions are drawn in Sect. 5.

2 Description of Proposed FOPID + FIR Controller

Proposed controller is based on elastic structure, which among the others, depends from number of controller input signals fb_i , $i = 1, \dots, FB$, FB stands for number of feedback signals (see Fig. 1). In proposed structure assumptions that fb_1 stands for desired value of fb_2 and the rest of the feedback signals stand for additional measurable signals are stated. Moreover, FOPID elements and FIR filters and their inner elements can be dynamically switched off or on by changing controller parameters. Due to that, the design of the controller should not only consider selecting the real parameters of the controller but also integer parameters encoding its structure. The typical FOPID control block consist of five elements: proportional P, integral I and λ and differential D and μ and its output is calculated as follows:

$$u(t) = K^P e(t) + K^I \left(\int_0^t e(t) dt \right)^{-\lambda} + K^D \left(\frac{de(t)}{dt} \right)^\mu, \quad (1)$$

where K^P , K^I and K^D stand respectively for parameters of P, I and D elements of control block, λ and μ are additional degrees of freedom in a comparison to typical PID controller structure, $e(t)$ stands for input of FOPID block. These parameters allow to handle higher order processes by performing optimization with various integral performance indices. The proposed elastic FOPID structure (noted as Control Block CB—see Fig. 2a) allows for additional reduction of P, I, D, λ and μ elements by using integer values C^P , C^I , C^D , C^λ , C^μ and reduction of whole control block by using integer value C^{CB} . The reduction takes place if the integer values are set to 0. Then, the output of the proposed FOPID takes the following form:

$$u(t) = \begin{cases} C^P K^P e(t) + C^I K^I \left(\int_0^t e(t) dt \right)^{ka(t)} + C^D K^D \left(\frac{de(t)}{dt} \right)^{kb(t)} & \text{for } C^{CB} = 1, \\ e(t) & \text{for } C^{CB} = 0 \end{cases}, \quad (2)$$

where $ka(t)$ stands for $-\lambda$ when $C^\lambda = 1$ and 1 when $C^\lambda = 0$, $kb(t)$ stands for μ when $C^\mu = 1$ and 1 when $C^\mu = 0$ (if $C^\lambda = 0$ and $C^\mu = 0$ proposed FOPID controller work

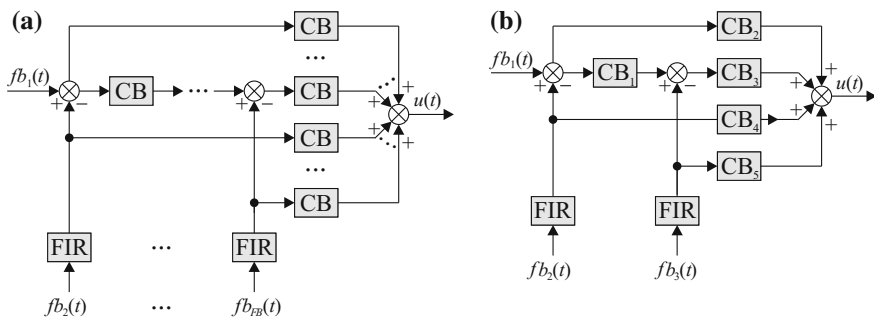


Fig. 1 Proposed controller structure: **a** with any number of FB feedback signals, **b** with 3 feedback signals

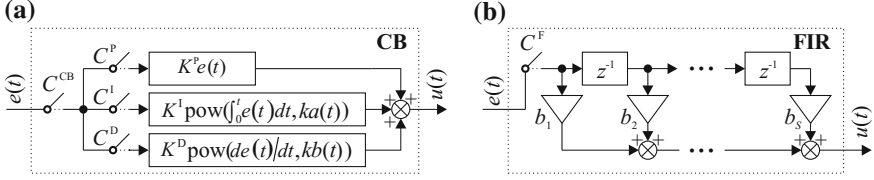


Fig. 2 Structure of: **a** proposed elastic control block CB based on FOPID, **b** proposed elastic filter FIR, z^{-1} stands for values from previous time step

as typical PID controller). The proposed FIR filters used in controller are based on typical FIR filters (see Fig. 2b) with using an additional integer parameter C^F standing for reduction of the filter. Thus, the output of the proposed filter takes the following form:

$$u(t) = \begin{cases} \sum_{s=1}^S b_s e(t-s-1) & \text{for } C^F = 1 \\ e(t) & \text{for } C^F = 0 \end{cases}, \quad (3)$$

where $e(t)$ stands for input value, $e(t-i)$ stands for input value from $t-i$ time step, b_s stands for weights of filter, $s = 1, \dots, S$, S stands for length of the filter (S has to be an odd number). The weights of the filter are calculated using filter parameters: transition frequency ft and length of the filter S , which are a part of the elastic structure of the controller and should be selected by learning algorithm as well. The weights values b_s are calculated as follows:

$$b_s = \begin{cases} \frac{\sin(2\pi ft |s - \frac{1}{2}(S-1)|)}{\pi |s - \frac{1}{2}(S-1)|} & \text{for } s = \frac{1}{2}(S-1) \\ 2ft & \text{for } s \neq \frac{1}{2}(S-1) \end{cases}. \quad (4)$$

The proposed controller structure is characterized by the following advantages: (a) possibilities of processing any number of feedback signals fb_i , (b) it uses cascade control blocks configuration which allows us to obtain good accuracy of the controller, (c) the structure is dynamic, each CB block elements (P, I, D, λ , μ) and filter FIR can be switched off or on, (d) it has great capabilities of learning due to many selectable parameters (e) it is able to minimize the impact of feedback signals noise by use of the FIR filters.

3 Description of the Proposed Hybrid Algorithm

In our paper a hybrid evolutionary algorithm is used to select the proposed controller parameters and structure. It is based on an ensemble of genetic algorithm (to select controller structure) and evolutionary strategy (to select controller

parameters). This ensemble was proposed in our previous work and it achieved good results. In this paper we propose a number of improvements that may allow us to obtain better performance. These improvements consider, most of all, iteration-dependent parameters of learning process and their description can be found in detail in the current section.

3.1 Encoding of the Controller Parameters

The parameters and the structure of the proposed controller are encoded in chromosome \mathbf{X}_{ch} defined as follows:

$$\mathbf{X}_{ch} = \{\mathbf{X}_{ch}^{\text{par}}, \mathbf{X}_{ch}^{\text{str}}\}, \quad (5)$$

where part $\mathbf{X}_{ch}^{\text{par}}$ encodes the real parameters of the controller and part $\mathbf{X}_{ch}^{\text{str}}$ encodes integer parameters of the controller. The part $\mathbf{X}_{ch}^{\text{par}}$ is defined as follows:

$$\mathbf{X}_{ch}^{\text{par}} = \left\{ \begin{array}{l} K_1^{\text{P}}, K_1^{\text{I}}, K_1^{\text{D}}, \lambda_1, \mu_1, \dots, \\ K_M^{\text{P}}, K_M^{\text{I}}, K_M^{\text{D}}, \lambda_M, \mu_M \\ ft_1, \dots, ft_R \end{array} \right\} = \{X_{ch,1}^{\text{par}}, \dots, X_{ch,L^{\text{par}}}^{\text{par}}\}, \quad (6)$$

where $K_m^{\text{P}} \in [0, 20]$, $K_m^{\text{I}} \in [0, 50]$, $K_m^{\text{D}} \in [0, 5]$, $\lambda_m \in [0.5, 2.0]$, $\mu_m \in [0.5, 2.0]$, stand for CB P, I, D, λ , μ parameters, $m = 1, \dots, M$, M stands for number of CB blocks, $ft_r \in [0.1, 0.5]$ stands for transition frequency, $r = 1, \dots, R$, $R = FB - 1$ stands for number of filters, $L^{\text{par}} = 5M + R$ stands for number of genes in part $\mathbf{X}_{ch}^{\text{par}}$. The part $\mathbf{X}_{ch}^{\text{str}}$ is defined as follows:

$$\mathbf{X}_{ch}^{\text{str}} = \left\{ \begin{array}{l} C_1^{\text{P}}, C_1^{\text{I}}, C_1^{\text{D}}, C_1^{\lambda}, C_1^{\mu}, \dots, \\ C_M^{\text{P}}, C_M^{\text{I}}, C_M^{\text{D}}, C_M^{\lambda}, C_M^{\mu}, \\ C_1^{\text{CB}}, \dots, C_M^{\text{CB}}, C_1^{\text{F}}, \dots, C_R^{\text{F}} \\ F_1, \dots, F_R \end{array} \right\} = \{X_{ch,1}^{\text{str}}, \dots, X_{ch,L^{\text{str}}}^{\text{str}}\}, \quad (7)$$

where $C_m^{\text{P}} \in \{0, 1\}$, $C_m^{\text{I}} \in \{0, 1\}$, $C_m^{\text{D}} \in \{0, 1\}$, $C_m^{\lambda} \in \{0, 1\}$, $C_m^{\mu} \in \{0, 1\}$ stand for activation of CB P, I, D, λ , μ elements (values equal to 1 stands for active element), $C_m^{\text{CB}} \in \{0, 1\}$ stands for activation of m th control block, $C_r^{\text{F}} \in \{0, 1\}$ stands for activation of r th filter (values equal to 1 stands for active element), $F_r \in \{0, \dots, 9\}$ stands for length of the filter (real length of the filter is calculated as $S_r = 5 + 2F_r$ to obtain at least 5-size long filters), $L^{\text{str}} = 6M + 2R$ stands for number of genes in part $\mathbf{X}_{ch}^{\text{str}}$.

3.2 Proposed Algorithm Description

Proposed algorithm is based on new iteration-dependent mutation and crossover from genetic algorithm and evolutionary strategy. The algorithm works according to the following steps

- **Step 1. Initialization.** In this step the value *iteration* is set to 0. Next the N individuals (each individual \mathbf{X}_{ch} represents controller encoded by chromosome (5) are randomly initialized and stored in population \mathbf{P} . The initialization of individuals' genes is realized as follows: $X_{ch,g}^{\text{par}} = U^g(\underline{X}_{ch,g}^{\text{par}}, \bar{X}_{ch,g}^{\text{par}})$, where $U^g(a, b)$ returns a random real value from the range $[a, b]$, $\underline{X}_{ch,g}^{\text{par}}$ and $\bar{X}_{ch,g}^{\text{par}}$ stand respectively for minims and maxims values of genes $X_{ch,g}^{\text{par}}$, $g = 1, \dots, L^{\text{par}}$, $X_{ch,h}^{\text{str}} = U^h(\underline{X}_{ch,h}^{\text{str}}, \bar{X}_{ch,h}^{\text{str}})$, where $U^h(a, b)$ returns random integer value from the range $[a, b]$. $\underline{X}_{ch,h}^{\text{str}}$ and $\bar{X}_{ch,h}^{\text{str}}$ stand respectively for minims and maxims values of genes $X_{ch,h}^{\text{str}}$, $h = 1, \dots, L^{\text{str}}$.
- **Step 2. Evaluation.** In this step each individual is evaluated by fitness function defined as follows:

$$\text{ff}(\mathbf{X}_{ch}) = \sum_{f=1}^F w_f \cdot \text{ffcom}_f(\mathbf{X}_{ch}), \quad (8)$$

where $\text{ffcom}_f(\mathbf{X}_{ch})$ stands for fitness function components which depend from simulation problem (see Sect. 4), w_f stands for weights of components, $f = 1, \dots, F$, F stands for number of fitness function components.

- **Step 3. Probabilities calculation.** In this step the value *iteration* is incremented. Next, the dynamic parameters for mutation and crossover are calculated as follows: individual mutation probability $p_1 = 0.10 + 0.20 \cdot \alpha$, gene mutation range $p_2 = 0.05 + 0.20 \cdot \alpha$, gene mutation probability $p_3 = 0.01 + 0.10 \cdot \alpha$, α stands for iteration-dependent value calculated as:

$$\alpha = 1 - \frac{\text{iteration}}{\text{iteration}^{\text{max}}}. \quad (9)$$

where $\text{iteration}^{\text{max}}$ stands for maximum number of algorithm iterations. The purpose of iteration dependent probabilities is to increase the possibilities of accurate exploration of space exploration by decreasing influence and range of mutation.

- **Step 4. Reproduction.** In this step a N new individuals are created and stored in population \mathbf{P}' . For each individual the condition $U^g(0, 1) < p_c$ is checked (where $p_c \in (0, 1)$ stands for crossover probability). If this condition is met, new individual is created as a result of crossover between two individuals selected by the roulette wheel method [7] from population \mathbf{P} . Otherwise, the individual is created as a result of cloning and mutating of one individual, which

is also selected by the roulette wheel method [7] from population \mathbf{P} . The mutation is performed according to Eq. (11) (see Step 5). The genes obtained from crossover are calculated as:

$$\begin{cases} X_{ch,g}^{\text{par}} = \begin{cases} X_{ch,g}^{\text{A,par}} & \text{for } U^g(0,1) < 0.5 & \text{and } U^g(0,1) < p_3 \\ X_{ch,g}^{\text{B,par}} & \text{for } U^g(0,1) \geq 0.5 & \text{and } U^g(0,1) < p_3 \\ X_{ch,g}^{\text{A,par}} + U^g(0,1) \cdot (X_{ch,g}^{\text{B,par}} - X_{ch,g}^{\text{A,par}}) & \text{for } U^g(0,1) \geq p_3 \end{cases} \\ X_{ch,h}^{\text{str}} = \begin{cases} X_{ch,h}^{\text{A,str}} & \text{for } U^g(0,1) < 0.5 & \text{and } U^g(0,1) < p_3 \\ X_{ch,h}^{\text{B,str}} & \text{for } U^g(0,1) \geq 0.5 & \text{and } U^g(0,1) < p_3 \\ X_{ch,h}^{\text{A,str}} + U^h(X_{ch,h}^{\text{A,str}}, X_{ch,h}^{\text{B,str}}) & \text{for } U^g(0,1) \geq p_3 \end{cases} \end{cases}, \quad (10)$$

where $X_{ch,g/h}^{\text{A,str/par}}$ and $X_{ch,g/h}^{\text{B,str/par}}$ stand respectively for genes from the first and second parent. The purpose of Eq. (10) is to increase chance to select gene values directly from parents or in the other case to select gene values between gene values of parents (if condition $U^g(0,1) \geq p_3$ is met).

- **Step 5. Mutation.** In this step genes of individuals from population \mathbf{P}' are mutated. For each individual the condition $U^g(0,1) < p_m$ is checked (where p_m stands for mutation probability). If this condition is met, genes of \mathbf{X}_{ch} are modified as follows:

$$\begin{cases} X_{ch,g}^{\text{par}} = \begin{cases} X_{ch,g}^{\text{par}} + U^g(-1,1) \cdot p_2 \cdot (\bar{X}_{ch,g}^{\text{par}} - \underline{X}_{ch,g}^{\text{par}}) & \text{for } U^g(0,1) < p_1 \\ X_{ch,g}^{\text{par}} & \text{for } U^g(0,1) \geq p_1 \end{cases} \\ X_{ch,h}^{\text{str}} = \begin{cases} X_{ch,h}^{\text{str}} + U^h(-1,1) & \text{for } U^g(0,1) < p_3 \\ X_{ch,h}^{\text{str}} & \text{for } U^g(0,1) \geq p_3 \end{cases} \end{cases}. \quad (11)$$

- **Step 6. Repair.** This step purpose is to repair (cut to specified ranged) gene values of individuals from population \mathbf{P}' , which is executed as follows:

$$\begin{cases} X_{ch,g}^{\text{par}} = \min\left(\bar{X}_{ch,g}^{\text{par}}, \max\left(\underline{X}_{ch,g}^{\text{par}}, X_{ch,g}^{\text{par}}\right)\right) \\ X_{ch,h}^{\text{str}} = \min\left(\bar{X}_{ch,h}^{\text{str}}, \max\left(\underline{X}_{ch,h}^{\text{str}}, X_{ch,h}^{\text{str}}\right)\right) \end{cases}. \quad (12)$$

- **Step 7. Evaluation.** In this step all individuals from population \mathbf{P}' are evaluated according to fitness function (8).
- **Step 8. Merging.** This step aim is to select the best N individuals from merged populations \mathbf{P} and \mathbf{P}' . Selected individuals replace population \mathbf{P} .
- **Step 9. Stopping condition.** In this step the stop condition is checked

- ($iteration \geq iteration^{\max}$). If this condition is met, algorithm stops and the best individual according to the fitness function value is presented. Otherwise, the algorithm goes back to Step 3.

4 Simulation Results

In our simulations a problem of designing controller structure and tuning parameters for double spring-mass-damp object was considered (see Fig. 3). More details about this model can be found in our previous paper [9]. Object parameters were set as follows: spring constant $k = 10 \text{ N/m}$, coefficient of friction $\mu = 0.5$, masses $m_1 = m_2 = 0.2 \text{ kg}$. Initial values of: s^1, v^1, s^2, v^2 (s stands for position, v stands for velocity) were set to zero, and s^* is a desired position of mass m_1 (see Fig. 3), simulation length T^{all} was set to 10 s, output signal of the controller was limited to the range $u \in (-2, +2)$, quantization resolution for the output signal of the controller and for the position sensor for s^1 and s^2 was set to 8 bit, noise level of feedback signals was set to 1 %, time step in the simulation was equal to $T = 0.1 \text{ ms}$, while interval between subsequent controller activations were set to 20 simulation steps, number of model iteration is calculated as $Z = T^{\text{all}}/T$. The feedback signals for the controller was chosen as: $fb_1 = s^*$, $fb_2 = s^1$, $fb_3 = s^2$.

4.1 Problem Evaluation

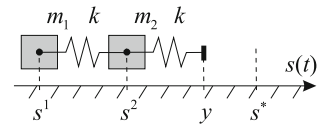
For problem under consideration a trapezoidal shape of desired signal s^* was used (see Fig. 4). Moreover, a following fitness function components (8) were used (additionally, a settling time can be included in a further research):

- Complexity of the controller:

$$\text{ffcom}_1(\mathbf{X}_{ch}) = \frac{1}{L^{\text{str}}} \sum_{g=1}^{L^{\text{str}}} \mathbf{X}_{ch,g}^{\text{str}}, \quad (13)$$

- *RMSE* standing for accuracy of the controlled object:

Fig. 3 Simulated spring-mass-damp object



$$\text{ffcom}_2(\mathbf{X}_{\text{ch}}) = \text{RMSE} = \sqrt{\frac{1}{Z} \cdot \sum_{i=1}^Z \varepsilon_i^2} = \sqrt{\frac{1}{Z} \cdot \sum_{i=1}^Z (s_i^* - s_i^1)^2}, \quad (14)$$

- Overshooting of the controller:

$$\text{ffcom}_3(\mathbf{X}_{\text{ch}}) = \max_{i=1, \dots, Z} \{s_i^1\}. \quad (15)$$

- Oscillations of the output of the controller:

$$\text{ffcom}_4(\mathbf{X}_{\text{ch}}) = \sum_{o=1}^{O-1} \sqrt{|r_o - r_{o+1}|}, \quad (16)$$

where r_o stands for each local minimis and maxims of the output values of the controller (minims and maxims were selected with ignoring noise influence on the signals), $o = 1, \dots, O$, O stands for number of minimis and maxims of oscillations. The aim of the Eq. (16) is to promote solutions with low number of low height oscillations of the controller.

4.2 Simulation Parameters

In the simulations the following values of parameters were set experimentally: fitness function components weights $w_1 = 0.1$, $w_2 = 10$, $w_3 = 0.01$, $w_4 = 0.1$, crossover probability $p_c = 0.75$, mutation probability $p_m = 0.75$, number of algorithm iterations $\text{iteration}^{\max} = 1000$, number of individuals in populations $N = 100$. In the simulations four cases presented in Table 1 were tested to show the effectiveness of the proposed controller and learning algorithm (case 1 corresponds to solution presented in [9]). For each case simulations were repeat 100 times and results were averaged.

4.3 Obtained Results

The averaged simulations results are presented in Table 2, the best simulations results are presented in Table 3, Figs. 4 and 5.

Table 1 Simulation cases

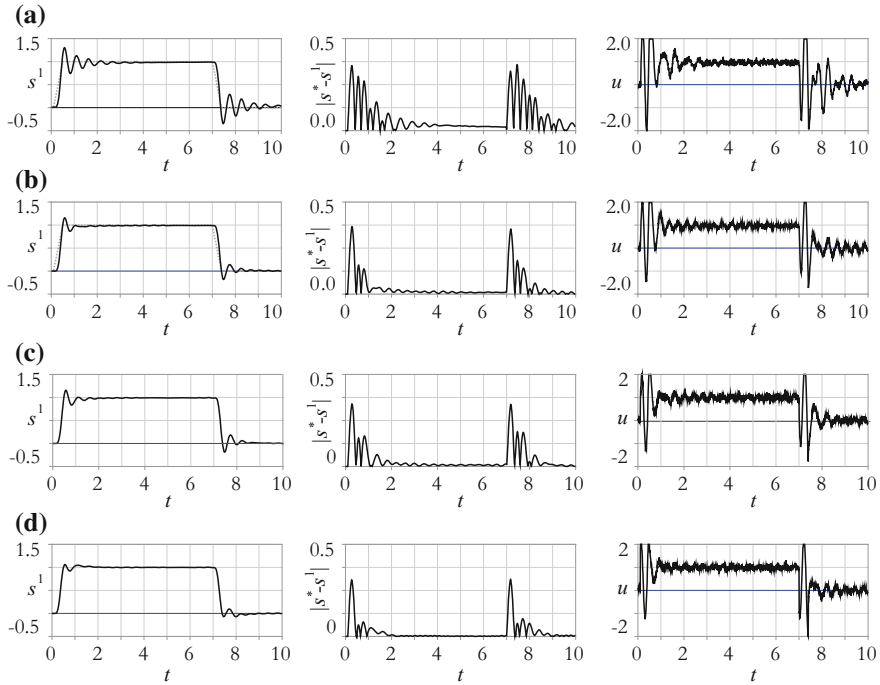
Case	Controller type	Filters	λ, μ always active
1	PID (FOPID with $\lambda = -1, \mu = 1$)	no	no
2	PID (FOPID + FIR with $\lambda = -1, \mu = 1$)	yes	no
3	proposed elastic FOPID + FIR	yes	no
4	proposed elastic FOPID + FIR	yes	yes

Table 2 Averaged simulation results

Case	ff(\cdot)	ffcom ₁ (\cdot) complexity	ffcom ₂ (\cdot) accuracy	ffcom ₃ (\cdot) oscillations	ffcom ₄ (\cdot) overshooting
1	3.628	0.437	0.162	31.743	1.190
2	3.080	0.502	0.144	13.767	1.160
3	2.382	0.640	0.114	14.468	1.195
4	2.373	0.664	0.107	27.651	1.221

Table 3 Best simulation results

Case	ff(\cdot)	ffcom ₁ (\cdot) complexity	ffcom ₂ (\cdot) accuracy	ffcom ₃ (\cdot) oscillations	ffcom ₄ (\cdot) overshooting
1	2.438	0.467	0.103	28.572	1.294
2	1.807	0.533	0.076	21.885	1.157
3	1.705	0.680	0.074	26.230	1.155
4	1.307	0.840	0.059	19.689	1.062

**Fig. 4** Best simulations results for: **a** case 1, **b** case 2, **c** case 3, **d** case 4. s^1 stands for position of the mass m_1 , $|s^* - s^1|$ stands for difference with desired position of mass m_1 , u stands for output of the controller

4.4 Simulation Results

Conclusions from the simulation are as follows: (a) adding filters allowed for reduce impact of feedback signals noise (see Fig. 4a), Fig. 4b) and $ff(\cdot)$ values in Table 2); (b) proposed elastic FOPID structure allowed for increase controller accuracy by around 20 % with increase of controller complexity by around 10 % (see Fig. 4b, c) and $ffcom_1(\cdot)$, $ffcom_2(\cdot)$ values in Table 2); (c) using static active λ , μ elements in FOPID structure allowed for next 5 % improvement in controller accuracy with slighter increase in controller complexity (see Fig. 4c, d) and $ffcom_1(\cdot)$, $ffcom_2(\cdot)$ values in Table 2); (d) the best obtained accuracy of the controller (see Table 3—case 4) is better than accuracy obtained by hybrid multi-population algorithms without noise of the signals under consideration (see our previous work [6]); (e) the obtained controller structures are simple and clear (see Fig. 5).

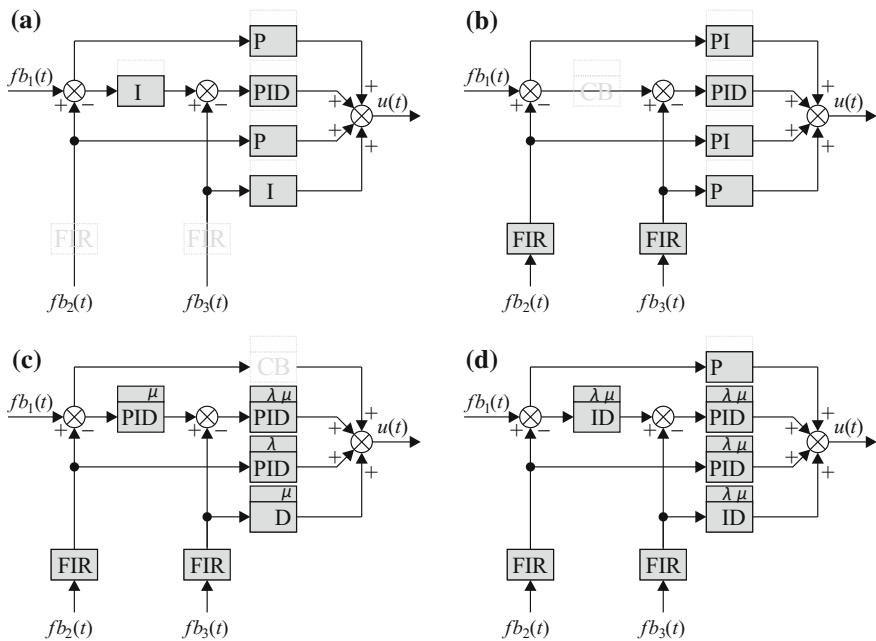


Fig. 5 Best simulations structures obtained for: **a** case 1, **b** case 2, **c** case 3, **d** case 4. Gray rectangles stands for reduced elements of the controller

5 Conclusions

In this paper a new method for elastic H_∞ -optimal fractional order PID with FIR filters (FOPID + FIR) controller design using hybrid population-based algorithm was proposed. The proposed elastic structure of the controller (FOPID + FIR) allowed to obtain overall good controllers (with good accuracy, small number of oscillations, low overshooting) with taking under consideration feedback signal noise and discretization. Moreover, the proposed training algorithm, which allows reduction of any component of the controller and simultaneously selection of its parameters, allowed to obtain a very good results in terms of accuracy. It can be said that the proposed elastic FOPID + FIR controller is superior in comparison to typical PID/FOPID controllers.

Acknowledgment The project was financed by the National Science Centre (Poland) on the basis of the decision number DEC-2012/05/B/ST7/02138.

References

1. Alia, M.A.K., Younes, T.M., Alsabbah, S.A.: A design of a PID self-tuning controller using LabVIEW. *J. Softw. Eng. Appl.* **4**, 161–171 (2011)
2. Astrom, K.J., Hagglund, T.: *PID Controllers: Theory, Design, and Tuning*. Instrument Society of America: Research Triangle Park, book (1995)
3. Cpałka, K., Łapa, K., Przybył, A.: A new approach to design of control systems using genetic programming. *Inf. Technol. Control* **44**(4), 433–442 (2015)
4. Ho, S.J., Ho, S.Y., Shu, L.S.: OSA: Orthogonal simulated annealing algorithm and its application to designing mixed $H_2 = H_1$ optimal controllers. *IEEE Trans. Syst. Man Cybern.* **34**(5), 588–600 (2004)
5. Łapa, K., Przybył, A., Cpałka, K.: A new approach to designing interpretable models of dynamic systems. *Lect. Notes Artif. Intell.* **7895**, 523–534 (2013)
6. Łapa, K., Szczypa, J., Venkatesan, R.: Aspects of structure and parameters selection of control systems using selected multi-population algorithms. *Artificial Intelligence and Soft Computing. Lecture Notes in Computer Science*, vol. 9120, pp. 247–260 (2015)
7. Rutkowski, L.: *Computational Intelligence*. Springer (2008)
8. Szczypa, J., Łapa, K., Shao, Z.: Aspects of the selection of the structure and parameters of controllers using selected population based algorithms. In: *Artificial Intelligence and Soft Computing*. Springer International Publishing, pp. 440–454 (2014)
9. Szczypa, J., Przybył, A., Cpałka, K.: Some aspects of evolutionary designing optimal controllers. *Artif. Intell. Soft Comput. part II* **7895**, 91–100 (2013)

Information Systems Architecture and Technology:
Proceedings of 37th International Conference on
Information Systems Architecture and Technology –
ISAT 2016 – Part II

Grzech, A.; Świątek, J.; Wilimowska, Z.; Borzemski, L.
(Eds.)

2017, XIV, 253 p. 115 illus., 28 illus. in color., Softcover
ISBN: 978-3-319-46585-2