SERBIAN JOURNAL OF ELECTRICAL ENGINEERING
Vol. 17, No. 1, February 2020, 65-82

UDC: 681.515.4:519.863 DOI: https://doi.org/10.2298/SJEE2001065B

Examples of Robust Controller Design

Anja Buljević[1], Miloš Miletić[2], Aleksandra Mitrović[1], Mirna N. Kapetina[1], Milan R. Rapaić[1]

Abstract: In this paper two modern controllers are presented. These controllers should eliminate disturbance effect, handle process variations and process uncertainties. First controller that is considered is conventional Proportional – Integral (PI) controller and second one is Fractional PI (FOPI) controller. After testing both controllers on a series of simulations, performance of both control algorithms were applied on a real 3D crane system.

Keywords: PI control, FOPI control, Particle Swarm Optimization (PSO), Symmetrical Optimum Method (SOM), Robust control.

1 Introduction

Conventional controllers of PI (and more generally, PID) type are among the most widely used solutions to practical control problems in industry, primarily due to their robustness simplicity, and well-established and widely-known tuning procedures. Practical controllers must exhibit satisfactory tracking performance, should be able to efficiently eliminate disturbances and be able to handle noisy inputs. In addition, controllers must be able to handle process variations, and process uncertainties in general. Of course, it is impossible to satisfy all these requirements in the same time, so it is necessary to find some compromise between them. In the present paper we illustrate two modern approaches for robust control synthesis. In the first one, we design a conventional PI controller, but tune the parameters in such a way to optimize a given performance index, while simultaneously constraining several performance-related and robustness indices. In the second one, we use a fractional PI controller, and try to exploit the additional degree of freedom introduced by the tunable order of integration. The main advantage of the first approach is that the resulting controller is of the conventional PI form, so it is straightforward to implement. The main advantage of the second procedure is that the tuning formulae are explicit and analytic, but the resulting controller is of fractional order and requires an additional effort in the implementation phase.

[1]University of Novi Sad, Faculty of Technical Sciences, Trg Dositeja Obradovića 6, 21000 Novi Sad, Serbia;
E-mails: anjabuljevic@uns.ac.rs; aleksandra.mitrovic@uns.ac.rs; mirna.kapetina@uns.ac.rs; rapaja@uns.ac.rs
[2]Typhoon HIL, Bajči Žilinskog bb, 21000 Novi Sad, Serbia; E-mail: m.miletic@uns.ac.rs

Both controllers are tested first on a series of simulations, and then experimentally, using laboratory pilot-plan of a 3D crane. In all cases, the nominal process model is assumed to be described by

$$G(s)=\frac{K}{s(Ts+1)}, \tag{1}$$

since this is an ubiquitous model, describing, among other things, various kinds of mechanical motion.

The paper is organized in the following manner: optimal design method for conventional PI controllers is introduced in Section 2, symmetrical optimum procedure for fractional order PI controllers is described in Section 3. The two proposed control algorithms were compared on model in Section 4, while description of the 3D crane system, process model and results on real system are given in Section 5, and the concluding remarks are given in the final Section 6.

2 Optimal PI Control Design

The topic of this section is determining optimal PI controller parameters for controlling second-order system. Main task is to determine parameters that satisfies some performance and robustness measures of the system. Parameters are optimized using PSO [1 – 3] algorithm. Controller must follow the reference, eliminate the disturbance and decrease noise impact. Also, controller must be resistant on the process variations, i.e. it should be as robust as possible in that sense. Although things mentioned above are very important in industry, the main requirement is to keep system stable. Of course, it is almost impossible to satisfy all these requirements in the same time, so it is necessary to find some compromise between them. Requirements are mathematically formulated by a suitably selected optimality criterion and accompanying constraints.

One of the most common system performance indicator is integral of absolute error (IAE) [4]. Measure of the system performance called Q [4 – 5] limits resonant peak of the frequency characteristic. By the limiting Q it is possible to get acceptable values of the IAE. Q is limited by 1.01 [4]. That value was obtained experimentally through a large number of simulations. This measure is defined by

$$Q=\max_{\omega\geq 0}\left|\frac{K_i\dfrac{G_p(\mathrm{j}\omega)}{\mathrm{j}\omega}}{1+G_c(\mathrm{j}\omega)G_p(\mathrm{j}\omega)}\right|,$$

where G_c is controller transfer function, G_p is process model transfer function, K_i is controller integral gain and ω corresponds to the frequency.

Also, it is necessary to define some systems robustness measures. In that purpose the most often used measure is maximal sensitivity M_s [4], defined by

$$M_s = \max_{\omega\geq 0}\left|\frac{1}{1+G_c(\mathrm{j}\omega)G_p(\mathrm{j}\omega)}\right|.$$

Measure M_s represents minimal inverse distance of the Nyquist curve from the critical point (-1, 0j). Often measure M_s is taken as the main systems robustness criterion. For example, system can have big gain and phase margin, but minimal distance of the Nyquist curve from the critical point can be small. Thus, if disturbance occurs at critical frequency, Nyquist curve can shift around critical point and system will become unstable. Best results are gained if maximal sensitivity is between 1.7 and 2 [4].

Maximal complementary sensitivity M_p [4]is another one system robustness measure which will be explained and used for the purposes of this paper. Application of the measure relates to examination of systems robustness in the medium frequency band [4]. It is used to get bigger damping or bigger phase margin without getting system response slower [4]. Maximal complementary sensitivity is defined by

$$M_p = \max_{\omega\geq 0}\left|\frac{G_c(\mathrm{j}\omega)G_p(\mathrm{j}\omega)}{1+G_c(\mathrm{j}\omega)G_p(\mathrm{j}\omega)}\right|.$$

The next step is to optimize the parameters of PI controller for each axis separately, i.e. mutual impact between axes is ignored in the optimization. Optimality criterion is the integral of absolute error (IAE)

$$\int_0^\infty |e(t)|\,\mathrm{d}t,$$

while constraints are Q (limit on resonant peak of frequency characteristic), maximal sensitivity and maximal complementary sensitivity

$$Q = \max_{\omega\geq 0}\left|\frac{K_i\dfrac{G_p(\mathrm{j}\omega)}{\mathrm{j}\omega}}{1+G_c(\mathrm{j}\omega)G_p(\mathrm{j}\omega)}\right| \leq Q^{\max}, \tag{2}$$

$$M_s^{\min} \leq \max_{\omega\geq 0}\left|\frac{1}{1+G_c(j\omega)G_p(j\omega)}\right| \leq M_s^{\max}, \tag{3}$$

$$M_p^{\min} \leq \max_{\omega\geq 0}\left|\frac{G_c(\mathrm{j}\omega)G_p(\mathrm{j}\omega)}{1+G_c(\mathrm{j}\omega)G_p(\mathrm{j}\omega)}\right| \leq M_p^{\max}. \tag{4}$$

Optimization is done by Particle Swarm Optimization (PSO) algorithm. This controller will be called Optimal PI. Obtained parameters for this controller and model responses will be shown in Section 4 and Section 5 where obtained responses will be compared with FOPI controller.

2.1 Optimization notes

As it is mentioned in pervious subsection, parameters are optimized using PSO algorithm. PSO was first time introduced by Kennedy and Eberhart in 1995 [3]. The idea was to derive mathematical model of the animals flock movement synchronization although there is no obvious communication between them. Due to this fact, PSO belongs to the group of evaluative algorithms, alongside with Genetic algorithm, Ant colony optimization etc.

Every single possible solution to the problem is called particle. Population of particles is called swarm. First step in the PSO algorithm is initialization. Swarm is initialized stochastically in the solution space. Every particle has potential solution to the problem and that vector is called position. Alongside with postion, every particle has velocity vector. Also, every particle can remember best position where it was in the past. Whole swarm remembers global best position. In each iterration, every particle calculates new velocity vector based on velocity vector from last iterration, personal best position and global best position. First velocity component, which is based on last velocity, is called inertial component. Second velocity component is called cognitive component and it presents particle tendency to move in the direction of its best attained position. Third velocity component, called social component, represents particle tendency to move in the direction of global best attained position. After calculating velocity vector, new particle position is calculated by simply adding velocity vector to current position. Algorithm is executed until maximum number of iterations is reached or when there is no progress.

Optimization constraints can be included in the criteria using penalty functions. So, extended optimization criteria can be written as follows

$$\int_0^\infty |e(t)|\,\mathrm{d}t + p_1\max(0, Q - Q^{\max}) + p_2\max(0, M_p - M_p^{\max}) +$$

$$+ p_3\max(0, -M_p + M_p^{\min}) + p_4\max(0, M_s - M_s^{\max}) + p_5(0, -M_s + M_s^{\min}),$$

where p_i are penalty functions coefficients which are usually set to be to some high values, for example 10 million. Population size is set to 30 particles and number of iterations is 150.

3 Fractional Order PI Control

Fractional controllers have been widely studied starting from the last decade of the 20th century [6 − 8]. Among the first widely known fractional order controllers was CRONE (Controle Robust de Ordre Non-Entier) proposed by Oustaloup and his team in Bordoux [22]. Fractional generalization of the conventional PID controller, obtained by allowing non-integer order differential and integral action, has been proposed by I. Podlubny in [6]

$$G_{pid_{f}o} = K_p + \frac{K_i}{s^{\alpha}} + K_d s^{\eta},$$

where K_p, K_i and K_d are the proportional, integral and derivative gain, respectively. More general fractional order controllers have been also considered, see for example [4], but in the present paper we focus on the fractional order PI controller

$$\mathrm{G}_{c_{f}o}(s) = K_p\left(1 + \frac{1}{(T_i s)^{\alpha}}\right), \quad 0 < \alpha < 1, \tag{5}$$

where, T_i is integral constant and it can be obtained using following formula

$$T_i^{\alpha} = \frac{K_p}{K_i}.$$

There are many methods for tuning parameters of PI controllers. Symmetrical Optimum Method (SOM) is one of the most popular and widely used. This concept was first presented by Kessler [9] in 1958. Later, in 1995 it was modified by Voda and Landau whose modified method ensures obtaining a maximum phase margin in closed loop system [11]. SOM has a few advantages over other methods in aspect of phase and gain margins and sensitivity of system.

In this paper, a generalisation of SOM applicable to fractional-order controllers is used. This method was introduced by Maione in [12]. The main principle is based on designing the regulator in order to meet requirements, so that phase margin is as close as it is possible to $37°$. That phase margin should be obtained at frequency $\omega_{\mathrm{max}} \equiv \omega_{\mathrm{gc}}$, where ω_{max} is the frequency at which maximum margin can be reached and ω_{gc} is the gain crossover frequency. Furthermore, $|\mathrm{W}_{sp}| = 1$ should be obtained in the largest possible frequency range, especially in low frequencies where W_{sp} is the transfer function of the closed-loop system. According to [12], this method also reduces disturbance effects.

SOM design is based on shaping asymptotic gain and choosing the slope of the segment crossing the frequency axis. The gain diagram should maintain this slope in a wide frequency interval around the crossover frequency, while the phase margin should be approximately constant in the same interval. In this manner, the phase margin is insensitive to gain variations, and robust stability is guaranteed.

If FO controllers are used, slopes can be $\alpha \cdot 20\mathrm{dB/dec}$ (α can be non-integer number, e.g. $\alpha = 0.5 \Rightarrow \mathrm{slope} = -10\mathrm{dB/dec}$) unlike classic controllers where α has to be integer number. Basic idea for tuning FO controllers parameters using SOM is to choose gain crossover frequency so that phase is maximized. Phase margin is maximum if

$$\frac{\mathrm{d}}{\mathrm{d}\omega}\arg \mathrm{W}(\mathrm{j}\omega) = 0 .$$

If it is not possible to achieve this, then it is possible to analytically solve non-linear minimization function to obtain the tuning formulae for controllers parameters. Comparing to the classical SOM, the additional tuning parameter α can be used to compromise between dynamic performance and robust stability. SOM tuning formula applicable for FOPI are derived in [12]

$$C = 1 + \cos(0.5\pi\alpha),\ S = \sin(0.5\pi\alpha),$$

$$\theta_a = \tan(0.5\pi - 0.5\pi\alpha),\ \theta_b = \tan(PM_\alpha),$$

$$a = \sqrt{\frac{C(1+\theta_a\theta_b) - S(\theta_a - \theta_b)}{C(\theta_a - \theta_b) + S(1+\theta_a\theta_b)}},$$

$$T_i = a^2 T,$$

$$K_p = \frac{1}{KT_i}\sqrt{\frac{1+a^4}{2a^4 C}} .$$

With this tuning obtained phase margin is

$$PM_\alpha = \arctan\frac{S}{C} - \arctan\left(a^{-2}\right) + 0.5\pi(1-\alpha).$$

3.1 Oustaloup-recursive-approximation for fractional order differentiators (ORA FOC)

Although many natural and artificial systems can theoretically be modeled and controlled successfully by fractional-order systems, there are difficulties in achieving analytic solutions for fractional differential equations. This is the biggest drawback of the use of fractional-order model. In addition, this also makes it tougher for simulation of fractional-order systems and implementation of fractional-order controllers. Therefore, a range of approximate methods for solving this problem have been proposed in recent years. As the basis of the numerical methods, the approximation of fractional-order operators plays a significant role among the studies of fractional order systems. There are many known approximations which can successfully describe fractional-order systems with the use of integer-order, such as Carlson's method [13], continuous fractional expansion method [14], Oustaloup's method [15], and many others.

Oustaloup recursive approximation (ORA) is probably the most famous and widely-used method used for this purpose. ORA is a fundamental tool used to find a rational integer-order approximation for fractional-order integrators of the form s^{α} expressed as

$$s^{\alpha}=k_f\prod_{n=1}^{N}\frac{1+\dfrac{s}{\mu_{z,n}}}{1+\dfrac{s}{\nu_{p,n}}}=G_{ORA}(s)\ ,$$

where k_f , *N*, $\mu_{z,n}$ and $\nu_{p,n}$ are unknown parameters that have to be determined. k_f denotes a steady-state gain, *N* denotes the order of approximation, $\mu_{z,n}$ and $\nu_{p,n}$ denote frequencies of poles and zeros and they can be calculated as follow

$$\mu_{z,1}=\omega_l\sqrt{\eta}\,,$$

$$\nu_{p,n}=\mu_n\gamma,\qquad n=1,\ldots,N\ ,$$

$$\mu_{z,n+1}=\nu_n\eta,\qquad n=1,\ldots,N-1\,,$$

$$\gamma=\left(\frac{\omega_h}{\omega_l}\right)^{\frac{\alpha}{N}},\qquad \eta=\left(\frac{\omega_h}{\omega_l}\right)^{\frac{1-\alpha}{N}}. \tag{6}$$

In (6) ω_l and ω_h describe the range of angular frequency, for which parameters are calculated. It is important to emphasize that the resulting approximation is only valid within a frequency range [ω_l , ω_h]. Implementation of this approximation can be found in [16].

4 Simulation Results

In the present section the two robust controllers, Optimal PI and FOPI, designed in the previous section are compared on a series of simulated situations.

The process model under consideration is given by (1). In simulations we use parameter values obtained by identifying a concrete physical crane model, to be described in the following section. Values of parameters of model that were used are shown in **Table 1**.

Table 1
Numerical values of unknown estimated parameters.

K_x	T_x	K_y	T_y
0.3075	0.1906	0.3176	0.1066

Now, transfer functions can be written as follows

$$G_x(s) = \frac{0.3075}{s(0.1906s+1)}, \tag{7}$$

$$G_y(s) = \frac{0.3176}{s(0.1066s+1)}, \tag{8}$$

where G_x is the model obtained for *x*–axis, and G_y is the model obtained for *y*–axis. The selected parameters are obtained by identification of a laboratory plant, as is discussed in the following section.

Design constraints and the resulting parameters of Optimal PI controller are shown in **Table 2** and **Table 3**, respectively. The first row in both tables contains values obtained for *x*–axis, while the second one contains values obtained for *y*–axis.

Table 2
Optimization constraints.

	$Q^{\max}$	$M_s^{\min}$	$M_s^{\max}$	$M_p^{\min}$	$M_p^{\max}$
x	1.01	1.3	1.5	1.2	1.4
y	1.01	1.4	1.6	1.2	1.4

Table 3
Obtained regulator parameters with corresponding constraints.

	K_p	K_i	Q	M_s	M_p
x	25.32	31.1	1	1.5	1.31
y	31.45	28	1	1.6	1.35

Design parameters for FOPI were selected on the basis of extensive numerical experimentation. It has been determined experimentally that satisfactory responses are obtained for $\alpha = 0.7$ with $N = 4$, $\omega_l = 0.1$ and $\omega_h = 100$. The obtained parameters for K_p and K_i are shown in **Table 4**.

Table 4

Obtained parameters for FOPI regulator where $\alpha = 0.7$.

	K_p	K_i
x	6.0389	11.9823
y	4.9848	9.7343

In all simulations, constant input disturbance of amplitude 0.3 starts acting at $t = 3$ s. In order to test robustness, some errors in modelling are also considered. Gain and time constant error simulates the situation in which process gain K and time constant T from system (1) are not correctly estimated. Results will be presented only for the x-axis because similar results are obtained for y–axis.

Fig. 3 shows model responses when only disturbance is included. It is obvious that Optimal PI faster eliminates disturbance impact and has smaller overshoot which is expected because Optimal PI controller's main task is to eliminate disturbance impact.

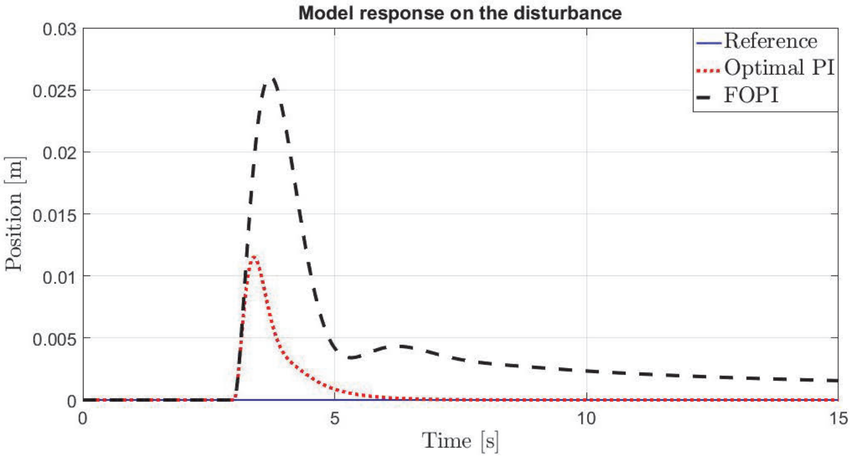


Fig. 3 – *Model responses on disturbance which starts after 3(s) with amplitude 0.3.*

Fig. 5a shows responses if the process gain is not estimated correctly. It was assumed that gain is 2, 4 and even 8 times bigger than real one and simulations for all of these assumptions are presented. It can be noticed that both controllers keep system stable and successfully eliminate disturbance because gain margin is big enough what can be seen at Fig. 4 which shows Bode diagrams of model controlled by FOPI and Optimal PI. By inspecting this figure, one may observe that Optimal PI manages to attain higher maximal phase compared to the FOPI controller, which naturally leads to higher maximal achievable phase margin. However, this maximum is achieved in a relatively narrow frequency band, which makes the phase margin highly sensitive to changes in the crossover frequency. Consequently, the phase margin of the Optimal PI controller is highly sensitive on gain variations. Practically, this means that, although Optimal PI controller is nominally more robust with respect to delay variations, it is relatively sensitive to combined variations in both delay and gain. Contrary to that, the maximal phase margin of the FOPI controller is smaller, but relatively high phase values are kept in a wide frequency band, making the phase margin relatively insensitive to process gain variations.

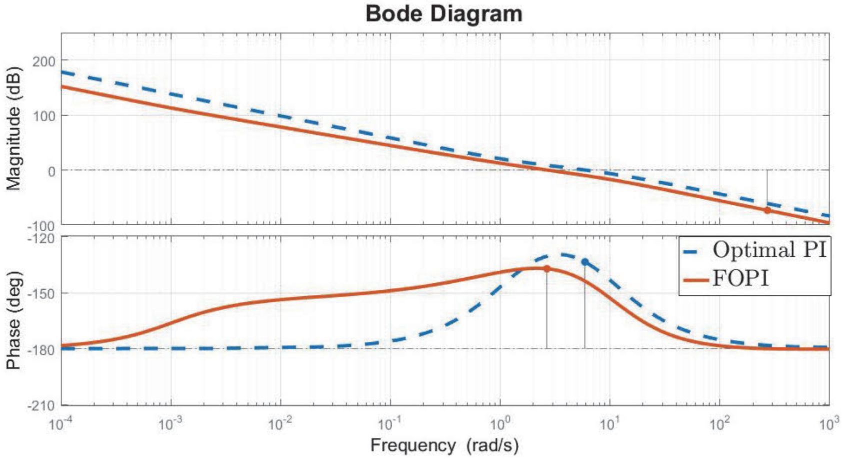


Fig. 4 – *Bode diagrams for model controlled by FOPI and Optimal PI. FOPI: gain margin is 73.3(dB), phase margin is 43°, delay margin is 0.282(s); Optimal PI: phase margin is 46.6 °, delay margin is 0.138(s).*

Fig. 5b shows responses if there is delay in system model. Following delays are included: 0.05s, 0.1s and 0.2s. It is obvious that FOPI controller is more robust in this aspect because it tolerates more delays. Optimal controller can tolerate delay for 0.1s, and after that system becomes unstable unlike FOPI controller which is still stable with same delay. Their delay margins are shown at Fig. 4 and they are 0.282s for FOPI controller and 0.138s for Optimal PI.

Fig. 5c shows responses when time constant is not estimated correctly. The experiments is performed with time constant 0.3 times smaller than real one, 1.5 times and 4 times bigger than real one, respectively. On figure 3c can be noticed that both controllers are very robust, but Optimal PI has better performance in this aspect.

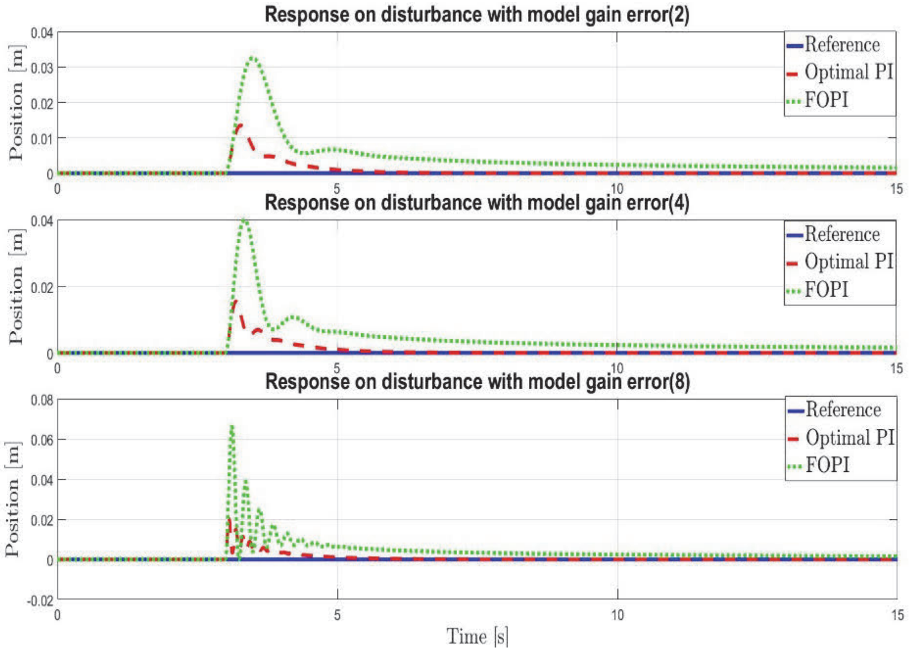


Fig. 5a – *Model responses with gain error.*

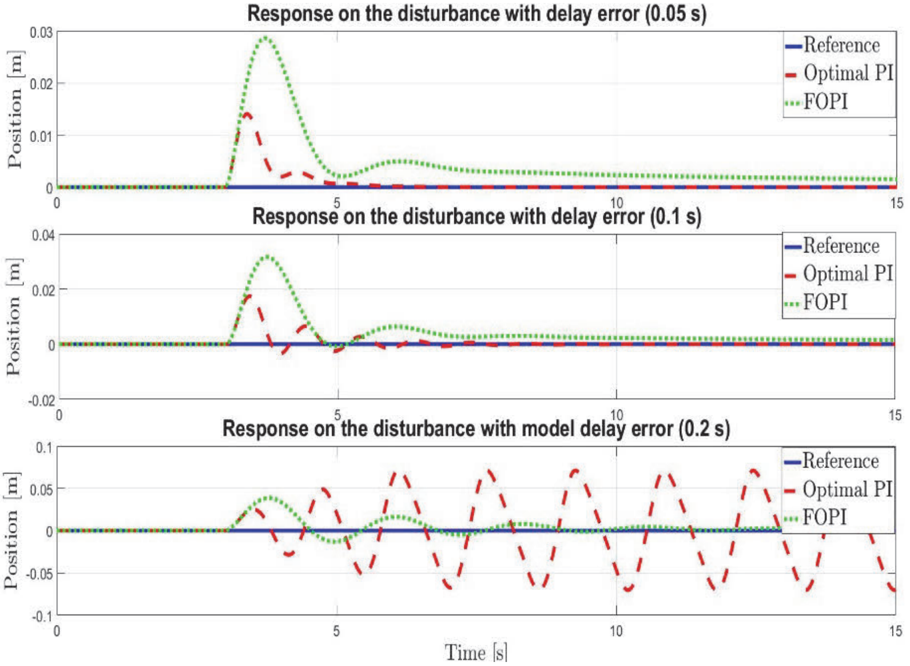


Fig. 5b – *Model responses with delay included in model.*

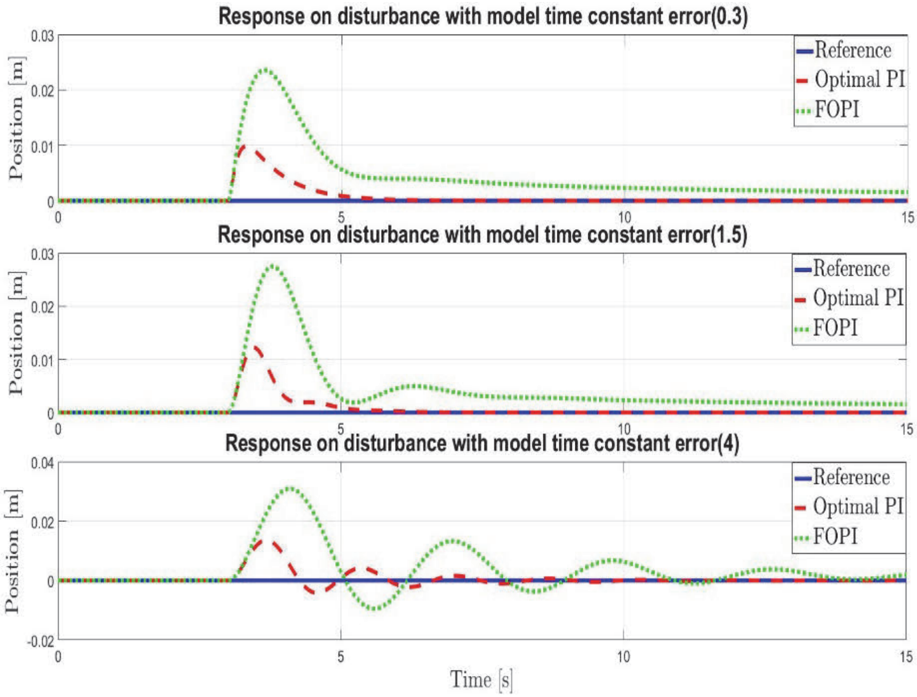


Fig. 5c – *Model responses when time constant is not estimated correctly.*

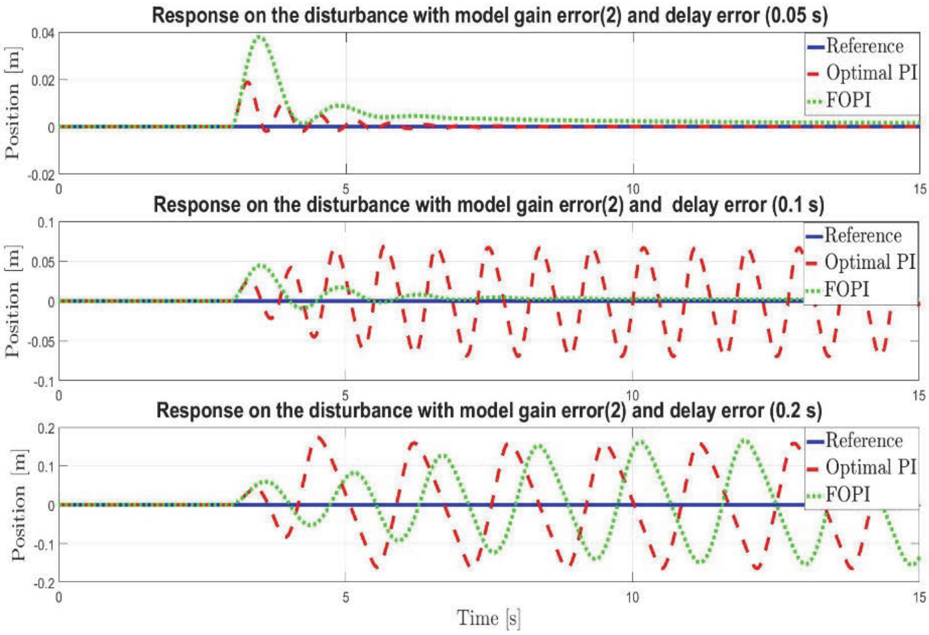


Fig. 5d – *Model responses with gain error and delay included.*

Fig. 5d shows responses when there is combination of wrongly estimated process gain and when there is a delay in system model. In all three simulations, gain is assumed to be 2 times bigger than real one and delays are 0.05s, 0.1s and 0.25s respectively. It is obvious that FOPI has better performance when few things are included in simulation, so we can say that FOPI controller is more robust than Optimal PI controller.

5 Robust Control of a 3D Crane System

In order to show the proposed controller algorithms are useful and robust enough, they were also applied on a real system which will be elaborated in the sequel. 3D crane is a real system on which control of Optimal PI and FOPI controller are tested.

Cranes are type of machine which purpose is to transfer heavy loads, beyond the normal capability of a man, from one place to another. Due to this fact, cranes are widely applied in industry. The main crane operations are hoist up-down motion and trolley forward-backward motion. These operations are controlled by human operators who use manual joysticks and mental map for control. However, the crane's operators sometimes need to complete their tasks in very narrow space for the short time. In that situation, when they need to react fast, operators could make error which may lead to operation failure. Furthermore, the load swing can damage the crane or the load or, even worse, it can cause human accidents. In order to avoid these unwanted situations, automated systems are implemented. Automated systems can help the operator to control the load swing or even replace the operator for having safe and efficient tower crane operation.

Inteco 3D crane [17] is laboratory model of 3D crane and it is controlled from a PC. The 3D crane is nonlinear electromechanical system having complex dynamic behavior and creating challenging control problems. Its hardware and software can be easily mounted and installed in a laboratory.

3D crane, shown on Fig. 6, consists of a payload hanging on pendulum which length can be changed. The payload can move freely in 3 dimensions (x, y and z–axis). Therefore, 3D crane is driven by 3 DC motors. The payload is lifted and lowered in the z direction. The rail and the cart are capable of horizontal motion along the rail in x direction. The cart is capable of horizontal motion along the rail in the y direction.

There are five identical measuring encoders measuring five state variables:

– the cart coordinates on the horizontal plane

– the lift-line length

– two deviation angles of the payload.

The encoders measure movements with a high resolution equal to 4096 pulses per rotation.

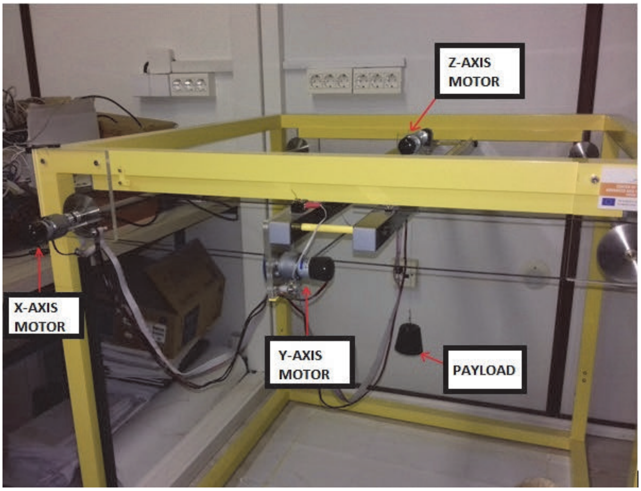


Fig. 6 – *3D Crane.*

5.1 Process model

In many papers 3D crane models are considered and various control techniques solving the position control to track desired trajectories and to reduce the load swing have been widely studied in the literature. Researchers in this area are making the significant effort in order to find the optimal and cheapest solution to control cranes, so it is possible to find many papers which describe different control algorithms. For instance, sliding-mode control method is used in [18]; Proportional – integer – derivative controller (PID) is used in [19] and partial feedback linearization control is used in [20].

Mathematical model of 3D crane can be described with ten nonlinear differential equations. In order to apply control design techniques of the previous sections, it is necessary to linearize the model. It is important to know that linear model obtained directly by least identification is of the same structure as adequate linearized nonlinear model.

$$G(s) = \frac{K_i}{s(T_i s + 1)}, \tag{9}$$

where K_i is gain, T_i is time constant, $i = x$ and $i = y$ for model which describes position of X – axis and Y – axis, respectively. Parameters K_i and T_i are unknown and will be estimated online.

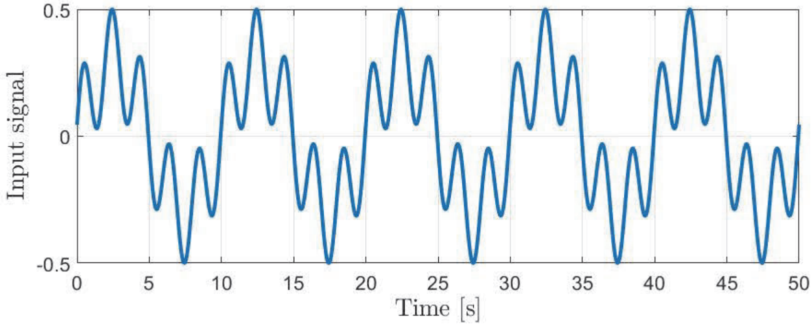
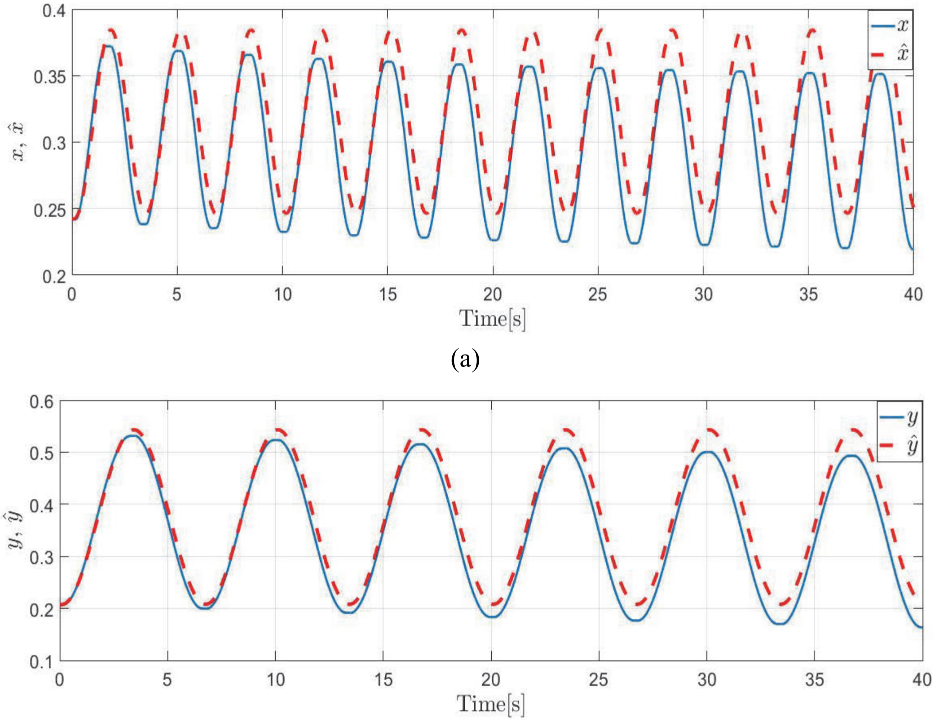


Fig. 7 – *Input signal.*



(b)

Fig. 8 – (a) *Comparative view of crane and model for x-axis.*
(b) *Comparative view of crane and model for y-axis.*

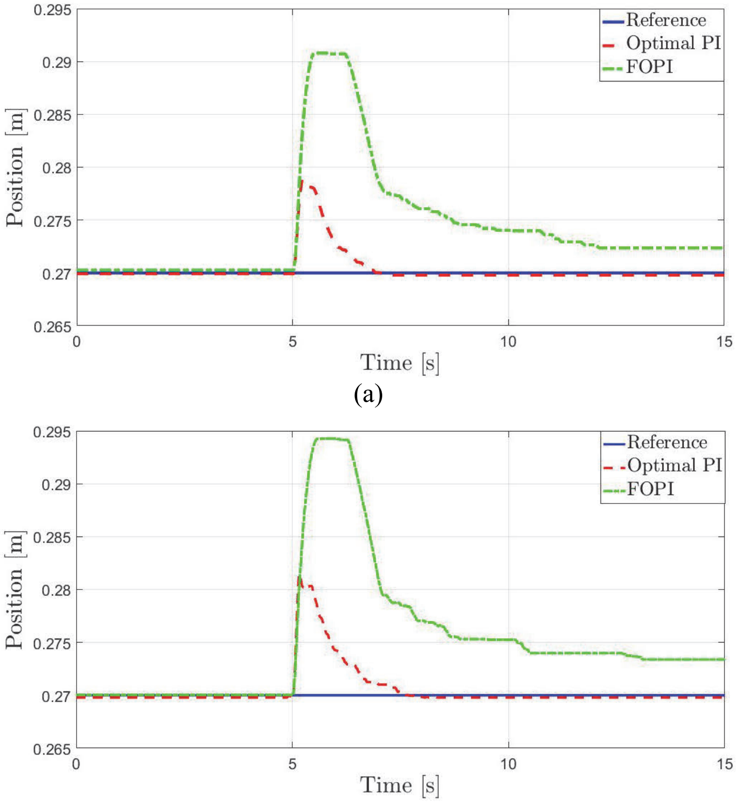
The least square method [21] was used for estimation of unknown parameters and their values for both axes are shown in **Table 1**. Input signal that was used is $u(t)=0.2\sin(\pi t+0.23)+0.3\sin(0.2\pi t)$ and it is shown at Fig. 7.

Comparative view between outputs of real system and model driven with the same input signal is shown at Fig. 8.

It can be noticed from Fig. 8 that due to the nonlinearity and imperfection of the system, there are disagreements between the linearized model and the real system. However, it will be shown that the linear model will be sufficient for robust control design.

5.2 Real system results

Tuning parameters for Optimal PI and FOPI controller, which are obtained using model (9), are given in the previous section. The same controllers are used on real 3D crane.



(a)

(b)

Fig. 9 – (a) *x-axis;* (b) *y-axis.*

Fig. 9 shows system responses when controllers should follow the reference and eliminate the disturbance. Disturbance is introduced as a step-like signal externally superimposed on the control signal, with amplitude equal to 30% of the maximal control signal value. Its effect begin in $t = 5$ s. Reference was set to be 0.27 m. Optimal PI controller shows better behavior because its parameters are purposely optimized to be robust on disturbance impact. FOPI controller should also eliminate disturbance impact but as it could be seen on figure it is not the case. It could be explained due to the fact that integrator of FOPI controller is not an integer order, so its effect is slower. When it gets into certain area (when steady-state error is very small), control force is growing very slowly and it has small value. Because of this, voltage is not big enough to start engine and that is the reason why FOPI has steady-state error. Theoretically, steady-state error will be zero after long enough time. On the other hand, FOPI is optimized to has bigger stability margins and it is more robust if we made some errors while modeling the system.

6 Conclusion

Two modern robust control synthesis methods were presented and compared in the present paper. The first one results in the conventional PI controller, and the parameter tuning is performed by maximizing performance taking into consideration robustness indices as constraints. The second one results in fractional order PI controller, and is obtained by means of analytical tuning formulae based on generalization of the well-known symmetrical optimum method. Both methods were implemented and tested both on a series of simulations, but also experimentally on a pilot plant of a 3D crane. It has been demonstrated that the optimal PI controller exhibits better tracking performances in general, but that FOP is more resilient when bigger delays are present in the control loop.

7 Acknowledgment

This work partially supported by Serbian Ministry of Education and Science, grant no.TR32018 (M.N.K., M.R.R.) and grant no. TR33013 (M.R.R.).

8 References

[1] Ž. Kanović, M. R. Rapaić, Z. D. Jeličić: Generalized Particle Swarm Optimization Algorithm - Theoretical and Empirical Analysis with Application in Fault Detection, Applied Mathematics and Computation, Vol. 217, No. 24, August 2011, pp. 10175 − 10186.

[2] M. R. Rapaić, Ž. Kanović: Time-Varying PSO-Convergence Analysis, Convergence-Related Parameterization and New Parameter Adjustment Schemes, Information Processing Letters, Vol. 109, No. 11, May 2009, pp. 548 − 552.

[3] J. Kennedy, R. Eberhart: Particle Swarm Optimization, Proceedings of the International Conference on Neural Networks (ICNN'95), Perth, Australia, November 1995, pp. 1942 − 1948.

[4] B. Jakovljević: Optimal and Suboptimal Parameter Tuning of Robust, Linear Controllers of Noninteger Order, Ph. D. Dissertation, Faculty of Technical Sciences Novi Sad, Novi Sad, Serbia, 2015. (in Serbian).

[5] T. B. Šekara, M. R. Mataušek: Optimization of PID Controller Based on Maximization of the Proportional Gain Under Constraints on Robustness and Sensitivity to Measurement Noise, IEEE Transactions on Automatic Control, Vol. 54, No. 1, January 2009, pp. 184 − 189.

[6] I. Podlubny: Fractional Order Systems and $PI^{\alpha}D^{\eta}$ Controllers, IEEE Transactions on Automatic Control, Vol. 44, No. 1, January 1999, pp. 208 − 214.

[7] M. R. Rapaić, T. B. Šekara: Novel Direct Optimal and Indirect Method for Discretization of Linear Fractional Systems, Electrical Engineering, Vol. 93, No. 2, January 2011, pp. 91 − 102.

[8] B. B. Jakovljević, T. B. Šekara, M. R. Rapaić, Z. D. Jeličić: On the Distributed Order PID Controller, AEU - International Journal of Electronics and Communications, Vol. 79, September 2017, pp. 94 − 101.

[9] C. Kessler: The Symmetrical Optimum, Part I, Regelungstechnik, Vol. 6, No. 11, November 1958, pp. 395 − 400. (in German).

[10] C. Kessler: The Symmetrical Optimum, Part II, Regelungstechnik, Vol. 6, No. 12, Decembar 1958, pp. 432 − 436. (in German).

[11] A. A, Voda, I. D. Landau: A Method for the Auto-Calibration of PID Controllers, Automatica, Vol. 31, No. 1, January 1995, pp. 41 − 53.

[12] G. Maione, P. Lino: New Tuning Rules for Fractional PI^{α} Controllers, Nonlinear Dynamics, Vol. 49, October 2006, pp. 251 − 257.

[13] G. E. Carlson: Simulation on the Fractional Derivative Operator $\sqrt{s}$ and the Fractional Integral Operator $1/\sqrt{s}$, MSc Thesis, Kansas State University, Manhattan, USA, 1960.

[14] S. D. Roy: On the Realization of a Constant-Argument Immittance or Fractional Operator, IEEE Transactions on Circuit Theory, Vol. 14, No. 3, September 1967, pp. 264 − 274.

[15] A. Oustaloup, F. Levron, B. Mathieu, F. M. Nanot: Frequency-Band Complex Noninteger Differentiator: Characterization and Synthesis, IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications, Vol. 47, No. 1, January 2000, pp. 25 − 39.

[16] Y. Q. Chen: Oustaloup-Recursive-Approximation for Fractional Order Differentiators, MATLAB Central File Exchange, Updated: August 2003, Available at:

https://www.mathworks.com/matlabcentral/fileexchange/3802-oustaloup-recursive-approximation-for-fractional-order-differentiators.

[17] 3DCrane User's Manual, Version 1.4, Inteco Ltd, 2008, Available at:

http://control.put.poznan.pl/old/sites/default/files/3DCrane.pdf

[18] D. Chwa: Sliding Mode Control-Based Robust Finite-Time Antisway Tracking Control of 3-D Overhead Cranes, IEEE Transactions on Industrial Electronics, Vol. 64, No. 8, August 2017, pp. 6775 − 6784.

[19] S. Y. S. Hussein, R. Ghazali, H. I. Jaafar, C. C. Soon: Analysis of 3D Gantry Crane System by PID and VSC for Positioning Trolley and Oscillation Reduction, Journal of Telecommunication, Electronic and Computer Engineering, Vol. 8, No. 7, October 2016, pp. 139 − 143.

[20] X. Wu, X. He: Partial Feedback Linearization Control for 3-D Underactuated Overhead Crane Systems, ISA Transactions, Vol. 65, November 2016, 361− 370.

[21] M. Kapetina: Adaptive Parameter Estimation in Systems Described by Irrational Transfer Functions, Ph. D. Dissertation, Faculty of Technical Sciences Novi Sad, Novi Sad, Serbia, 2017. (in Serbian).

[22] P. Lanusse, R. Malti, P. Melchior: CRONE Control System Design Toolbox for the Control Engineering Community: Tutorial and Case Study, Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences, Vol. 371, No. 1990, May 2013, pp. 1 − 14.