Central European Journal of Physics

Tuning fractional PID controllers for a Steward platform based on frequency domain and articial intelligence methods

--Manuscript Draft--

Manuscript Number:	
Full Title:	Tuning fractional PID controllers for a Steward platform based on frequency domain and articial intelligence methods
Article Type:	Topical Issue: Research Article
Section/Category:	Computational Physics
Keywords:	Steward platform, fractional-order, PID control, genetic algorithm, frequency domain
Corresponding Author:	Cosmin Copot Ghent University BELGIUM
Corresponding Author Secondary Information:	
Corresponding Author's Institution:	Ghent University
Corresponding Author's Secondary Institution:	
First Author:	Cosmin Copot
First Author Secondary Information:	
Order of Authors:	Cosmin Copot
	Yu Zhong
	Clara-Mihaela Ionescu, Dr.
	Robin De Keyser, Prof.
Order of Authors Secondary Information:	
Abstract:	In this paper, two methods to tune a fractional-order PID controller for a mechatronic system are presented. The first method is based on a genetic algorithm to obtain the parameter values for the fractional-order PID controller by global optimization. The second method used to design the fractional-order PID controller relies on an auto-tuning approach by meeting some specications in the frequency domain. The real-time experiments are conducted using a Steward platform which consists of a table tilted by six servo-motors with a ball on the top of the table. The considered system is a 6 degrees of freedom (d.o.f.) motion platform. The feedback on the position of the ball is obtained from images acquired by a visual sensor mounted above the platform. The fractional-order controllers were implemented and the performances of the steward platform are analyzed.
Suggested Reviewers:	Jose Antonio Tenreiro Machado jtm@isep.ipp.pt
	Cristina Muresan cristina.pop@aut.utcluj.ro
	Roxana Both roxana.both@aut.utcluj.ro
Opposed Reviewers:	

Cent. Eur. J. Phys. • 1-18
Author version

Central European Journal of Physics

Tuning fractional PID controllers for a Steward platform based on frequency domain and artificial intelligence methods

Article for Special Issue

Cosmin Copot [*], **Yu Zhong** [†], **Clara M. Ionescu,** [‡] **Robin De Keyser** [§]

Department of Electrical energy, Systems and Automation, Ghent University,
Technologiepark 913, 9052 Gent, Belgium

Abstract: In this paper, two methods to tune a fractional-order $PI^{\lambda}D^{\mu}$ controller for a mechatronic system are presented. The first method is based on a genetic algorithm to obtain the parameter values for the fractional-order $PI^{\lambda}D^{\mu}$ controller by global optimization. The second method used to design the fractional-order $PI^{\lambda}D^{\mu}$ controller relies on an auto-tuning approach by meeting some specifications in the frequency domain. The real-time experiments are conducted using a Steward platform which consists of a table tilted by six servo-motors with a ball on the top of the table. The considered system is a 6 degrees of freedom (d.o.f.) motion platform. The feedback on the position of the ball is obtained from images acquired by a visual sensor mounted above the platform. The fractional-order controllers were implemented and the performances of the steward platform are analyzed.

PACS (2008): 02.30.Yy, 45.10.Hj, 43.60.Uv

Keywords: Steward platform • fractional-order • PID control • genetic algorithm • frequency domain

1. Introduction

The mechatronic systems represents one of the most challenging control applications due to their interdisciplinary nature [1–3]. Numerous control algorithms have been proposed to deal with nonlinear dynamics of the mechatronic systems. For linear mechatronic systems, the proportional-integral-derivative (PID) controller is often used owing to its simple structure and robustness [4]. The advantage of the PID controller is that it can be easily tuned in a wide range of operating conditions [4, 5]. However, the design of conventional integer-order PID controllers should be based on the model.

[*] *E-mail: Cosmin.Copot@UGent.be*
[†] *E-mail: Yu.Zhong@UGent.be*
[‡] *E-mail: ClaraMihaela.Ionescu@UGent.be*
[§] *E-mail: Robain.DeKeyser@UGent.be*

Another approach in dealing with mechatronic systems challenges is the fractional-order (FO) control strategies. The concept of FO or non-integer order systems refers to those dynamical systems whose model structure contains arbitrary order derivatives and/or integrals [6–8]. In recent years, many studies and applications of fractional-order systems in areas such as science and engineering have been presented, but there is still much room for developing these emerging tools. With the development of the fractional-calculus, researchers in the engineering field realized that many real processes are described more adequately by fractional-order state equations [9–11]. Thus, the conventional integer-order PID controller becomes less suitable for the control of the fractional-order reality. A suitable way to improve the control performance is to use a controller with a similar structure as in natural world [12], i.e. a fractional-order $PI^{\lambda}D^{\mu}$ controller.

Recently, fractional-order control of nonlinear systems has started to attract the interest for applications in control engineering [13, 14]. By expanding derivative and integral terms to fractional-order in the controller, we can adjust the control system's frequency response directly and continuously. Fractional-order PID controllers have been used in industrial applications [8] and various areas such as mechatronic systems [15, 16], and bio-medical systems [17–19].

In this paper, two design methods of a fractional-order $PI^{\lambda}D^{\mu}$ controller for a mechatronic system are presented. The first method employs a genetic algorithm as a global optimization method to obtain the five parameters for the fractional-order controller in time domain. The benefit of this method is that it requires less knowledge about the process. The second method is numerical and based on an auto-tuning strategy. In order to analyze the performances of the controllers, a typical mechatronic system composed of a Steward platform with 6 degrees of freedom (d.o.f.) is employed. To control the position of the ball on the platform, a visual feedback is necessary. This feedback is given by a camera mounted on the top of the platform, while the Hough transform algorithm is used to detect the center of the ball. The experimental results reveal good performances and show a stable and convergent behavior of the Steward platform when dealing with fractional-order control. Both fractional-order controllers are implemented, tested and validated and their performances are analyzed.

The paper is organized as follows: in Section 2 the Steward platform (i.e. ball and plate system), its inverse kinematics, theoretical model, the visual feedback and the system identification and validation of the theoretical model are presented. In Section 3, we briefly introduce the integer-order and the fractional-order controllers, followed by two tuning methods for the fractional-order controllers: i) based on artificial intelligence and ii) based on frequency domain. The results of tuning, along with some implementation aspects followed by the experimental outcomes are presented in Section 4. The conclusions are detailed in the last Section.

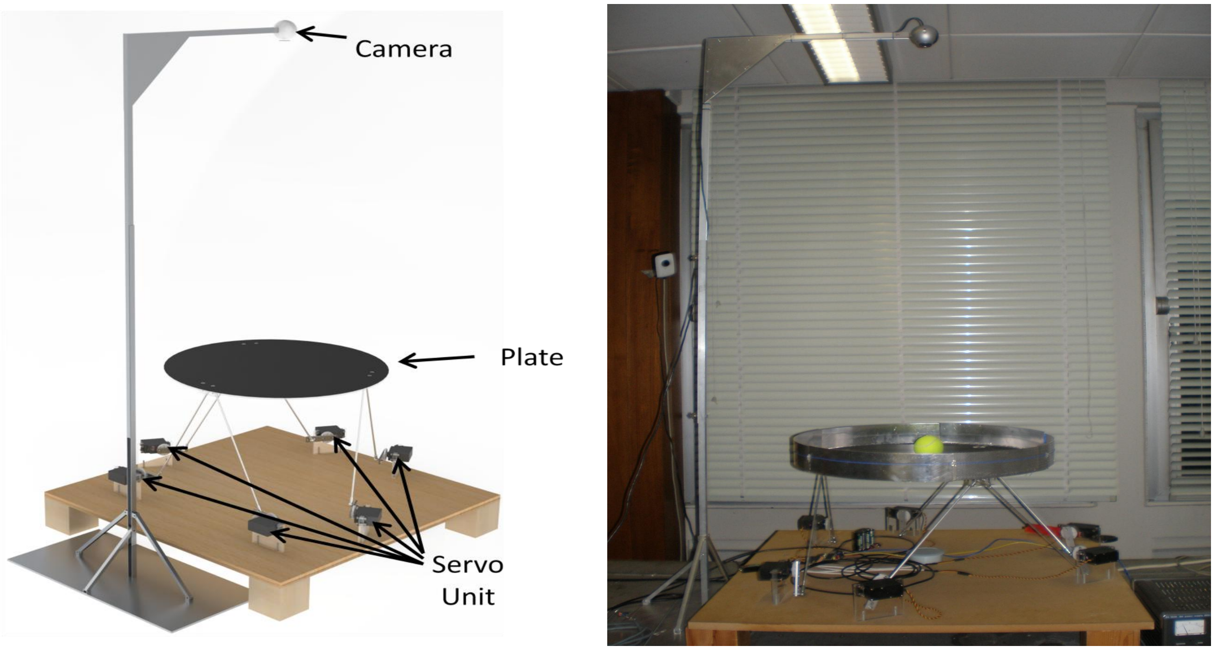
2. Steward Platform (ball and plate system)

The Steward platform consists of a table which is sloped by servo or electric motors and a ball that roll freely on the table. This ball and plate system can be considered as the two-dimensional extension of the ball and beam system.

2.1. Hardware design

In order to move the ball, the plate is tilted using 6 servo-motors and thus the platform has 6 d.o.f. in which a freely-suspended body can move. Three linear movements on x, y, z axis (lateral, longitudinal and vertical), and three rotations (pitch, roll and yaw) are available, but in this case the main movements for the ball and plate system are on the pitch and roll rotations. The remaining degrees of freedom can be exploited to optimize the control but this falls out of the scope of this paper.

The motion platform is called a Stewart platform [20]. This is a type of parallel robot that incorporates six actuators which are mounted in pairs to the base, crossing over to three mounting points on a top plate (Figure 1). R/C servos are devices, typically employed in radio controlled models, where they are used to provide actuation for various mechanical systems. A servomechanism is an automatic device that uses error-sensing negative feedback to control mechanical parameters, in this case position. Due to their affordability, reliability, and simplicity of control by microprocessors, R/C servos are often used in small-scale robot applications. R/C servos are composed of an electric motor mechanically linked to a potentiometer. A controller sends pulse-width modulation (PWM) signals to the servo with a 50 Hz frame rate. The electronics inside the servo translate the width of the pulse into a position. A downside is that there is no feedback from the servo. The microcontroller is unaware of the effective position of the servo.



(a)

(b)

Figure 1. Six DoF Motion Platform with camera for vision feedback; (a) the schematic representation; (b) the real system

2.2. Inverse kinematics

Based on the theoretical developments from [21], the inverse kinematic model is designed. The analytical method is derived in order to compute the position of the servo-motors for a given platform position. Each servo-motor has a local coordinate system defined as in Figure 2. The center of a servo-motor is the point A with coordinates (x_a, y_a, z_a). Let r be the length of the servo arm (AB) while BC is the shaft that connects the servo arm with the platform in the point C and has the length ℓ. As the position of the platform is known, the position of each connection (C) is also known (specified by the controller). The position of the servo (A) and its orientation (θ) are determined by the motion platform construction. The absolute distance between the point A and C is denoted by ℓ_1. The angle of the servo-motor relative to its neutral position (the angle of the servo arm with the xy-plane), is indicated by φ ($\varphi = 0^\circ$ in neutral position).

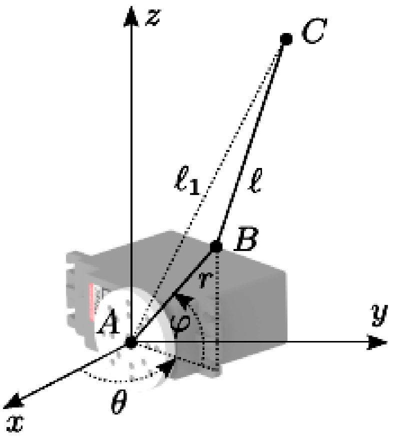


Figure 2. **Local coordinate frame for a servo control unit**

The output of the inverse kinematics is:

$$\varphi = \pm \arccos\left(\frac{M}{\sqrt{K^2 + H^2}}\right) + \arctan\left(\frac{H}{K}\right) + k\pi,\ \forall k \in \mathbb{Z}, \tag{1}$$

where:

$$\begin{aligned} M &= \ell^2 - \ell_1^2 - r^2 \\ K &= 2\left(x_a - x_c\right) r \cos\theta + x_a + 2\left(y_a - y_c\right) r \sin\theta + y_a \ \cdot \\ H &= 2\left(z_a - z_c\right) r \end{aligned} \tag{2}$$

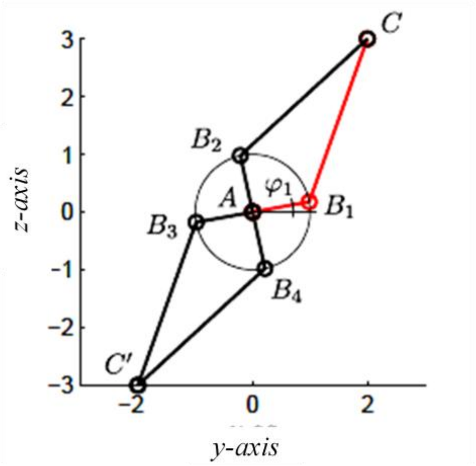


Figure 3. **Mapping of possible solutions for (1)**

As can be observed, equation (1) has four solutions:

$$\begin{aligned} \varphi_1 &= \arccos\left(\frac{M}{\sqrt{K^2+H^2}}\right) + \arctan\left(\frac{H}{K}\right) + \pi \\ \varphi_2 &= -\arccos\left(\frac{M}{\sqrt{K^2+H^2}}\right) + \arctan\left(\frac{H}{K}\right) + \pi \\ \varphi_3 &= \arccos\left(\frac{M}{\sqrt{K^2+H^2}}\right) + \arctan\left(\frac{H}{K}\right) \\ \varphi_4 &= -\arccos\left(\frac{M}{\sqrt{K^2+H^2}}\right) + \arctan\left(\frac{H}{K}\right) \end{aligned} . \tag{3}$$

All possible solutions are shown in Figure 3. By using mathematical methods, a second point C' is found, so solutions 3 (φ_3) and 4 (φ_4) are discarded. Because the servo-motor can move only in the interval $[-90°, 90°]$, the solution 1 (φ_1) will be selected.

2.3. Modeling

The motion of the ball on the plate can be described by using the Euler-Lagrange equation:

$$\frac{d}{dt}\frac{\partial L}{\partial \dot{\mathbf{q}}} - \frac{\partial L}{\partial \mathbf{q}} = Q, \tag{4}$$

where L is the Lagrangian and represents the difference between kinetic energy T and potential energy V and $\mathbf{q} = [x, y]^T$ is the vector of generalized coordinates. These are the coordinates of the ball on the platform (Figure 4).

The kinetic energy is the sum of the translation of the center of the ball and the rotation of the ball about its center. The rotation of the ball as a result of the rotation of the platform is not taken into account. The kinetic energy can be computed using:

$$T = \frac{1}{2}m\left(\dot{x}^2 + \dot{y}^2\right) + \frac{1}{2}J\left(\dot{\omega}_x^2 + \dot{\omega}_y^2\right) = \frac{1}{2}\left(m + \frac{J}{r_b^2}\right)\left(\dot{x}^2 + \dot{y}^2\right), \tag{5}$$

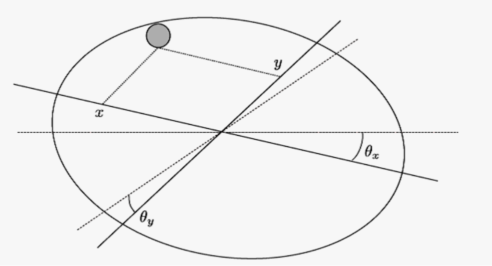


Figure 4. **Schematic drawing of the ball and plate arrangement**

where r_b is the radius of the ball, ω_x and ω_y represents the angular velocity of the ball related with x and y directions, m is the mass of the ball and J is the moment of inertia.

The potential energy can be written as following:

$$V = mgh = mg\left(x\sin\theta_x + y\sin\theta_y\right), \tag{6}$$

with g the gravitational acceleration, h the height of the ball with respect to a reference level and θ_x and θ_y the rotations of the platform along the x-axis and y-axis.

Inserting $\mathbf{q} = [x, y]^T$ in the Euler-Lagrange equation (4) and if we substitute the Lagrangian L, the following dynamic model is obtained:

$$\begin{cases} \left(m + \frac{J}{r_b^2}\right)\ddot{x} + mg\sin\theta_x = 0 \\ \left(m + \frac{J}{r_b^2}\right)\ddot{y} + mg\sin\theta_y = 0 \end{cases} \tag{7}$$

Taking into account that for a sufficiently small angle θ_x or θ_y the term $\sin\theta_x$ and $\sin\theta_y$ can be replaced with θ_x and θ_y, the linearization of the model can be written as:

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u} \tag{8}$$

where

$$\mathbf{x} = \begin{bmatrix} x \\ \dot{x} \\ y \\ \dot{y} \end{bmatrix}; \quad \mathbf{u} = \begin{bmatrix} \theta_x \\ \theta_y \end{bmatrix}; \quad \mathbf{A} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix}; \quad \mathbf{B} = -\frac{mg}{m + \frac{J}{r^2}} \begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 0 \\ 0 & 1 \end{bmatrix}. \tag{9}$$

Replacing the inertial term $J = \frac{5}{2}mr^2$, we can write the matrix $\mathbf{B}$ as:

$$\mathbf{B} = -\frac{5}{7}g \begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 0 \\ 0 & 1 \end{bmatrix}. \tag{10}$$

2.4. Visual feedback

In order to control the ball, a feedback of the position of the ball is necessary. This feedback is obtained using a visual sensor mounted on the top of the platform (Figure 1). This camera captures frames at 20Hz. The frames are interpreted by a self-written vision algorithm which, after calibration, determines the position of the ball on the platform.

Based on the Hough transform, which can be used to isolate features of a particular shape within an image, the center of the ball can be detected. The position of the ball in the image plane is assigned to a position on the platform using the pinhole camera model which is a mathematical relationship between the coordinates of a 3D point and its projection onto the image plane. Let $\mathbf{P} = [x_{\mathbf{P}}, y_{\mathbf{P}}, z_{\mathbf{P}}]^T$ be a point in Cartesian space, its projection in the image plane is the point $\mathbf{P'} = [u_{\mathbf{P'}}, v_{\mathbf{P'}}]^T$ expressed in pixels:

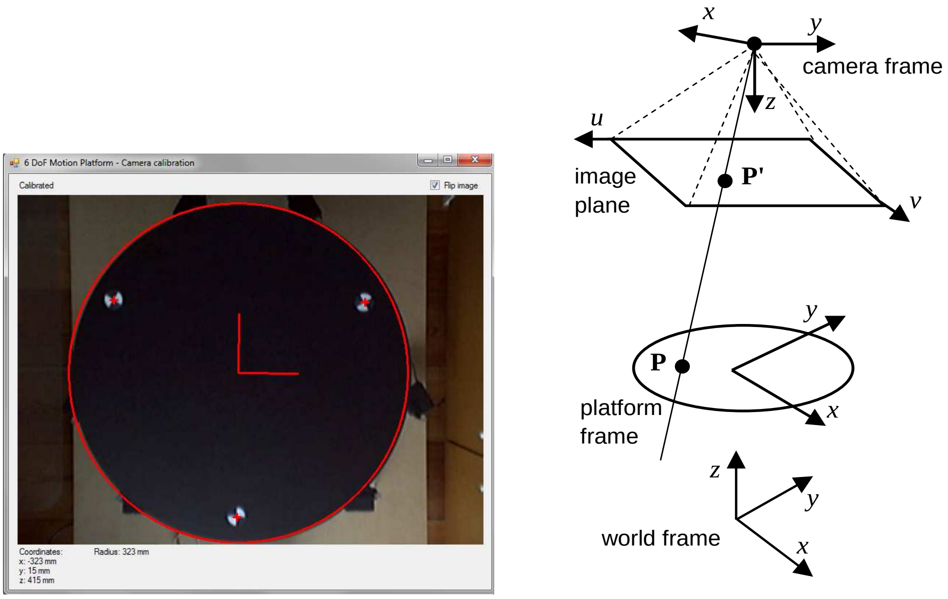
$$\begin{bmatrix} u_{\mathbf{P'}} \\ v_{\mathbf{P'}} \\ 1 \end{bmatrix} = \Upsilon \Pi_0 \begin{bmatrix} x_{\mathbf{P}} \\ y_{\mathbf{P}} \\ z_{\mathbf{P}} \\ 1 \end{bmatrix} \tag{11}$$

where:

$$\Upsilon = \begin{bmatrix} ls_x & ls_\theta & o_x \\ 0 & ls_y & o_y \\ 0 & 0 & 1 \end{bmatrix}, \Pi_0 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}. \tag{12}$$

In (12) Υ represents the matrix of the intrinsic parameters, l is the focal length, s_x, s_y are the scalar factors on x and y directions and o_x, o_y are the coordinates of the center of the image expressed in pixels. The intrinsic camera parameters are determined using the Camera Calibration Toolbox from Matlab. The extrinsic parameters describe the position of the camera with respect to the world frame. An algorithm was developed to determine the extrinsic parameters by selecting three markers in the camera picture, which are placed on well known positions on the platform (Figure 5(a)). Each time the position of the camera is changed, a new calibration should be preformed.

The algorithm that assigns a pixel in a camera frame to a position in Cartesian space is illustrated in Figure 5(b). Knowing the position of the camera in the world frame (extrinsic parameters) and the position of the image projection plane with respect to the optical center (intrinsic parameter), the light beams related to each pixel can be reconstructed. Taking into account that the position of the platform in the world frame is also known, it can be easily seen that the intersection of the light ray with the platform is in accordance with the pixel in the camera frame.


6 DoF Motion Platform - Camera calibration

Calibrated

☑ Flip image

Coordinates:
x: -323 mm
y: 15 mm
z: 415 mm

Radius: 323 mm

(a)

(b)

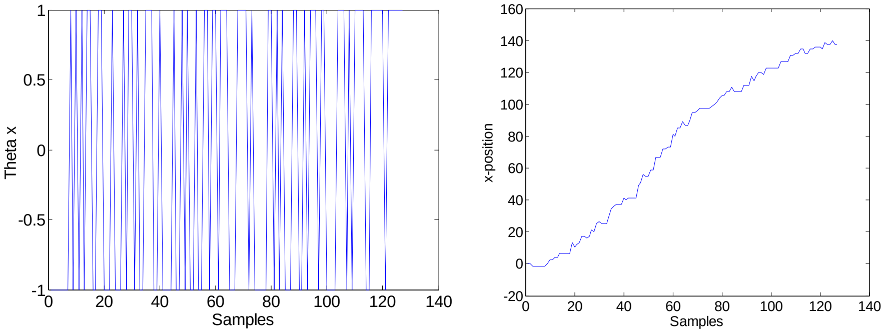
Figure 5. (a) Calibration of the extrinsic parameters with tree markers; (b) Assigning a pixel to a 3D position on the platform

2.5. Model validation

The signal used for the identification is a PRBS (pseudo random binary signal) signal. The length of the signal is $2^7 - 1 = 127$. To generate this PRBS signal, the following command is used in Matlab:

- idinput(127, 'PRBS', [0 1/1], [-1 1]).

This command results in the following signal (Figure 6(a)), and the system response is shown in Figure 6(b).



(a)

(b)

Figure 6. **(a) PRBS signal for identification; (b) Output of the x-position of the ball.**

By using the Prediction Error Method (PEM) for identification [22], the system's model is defined as:

$$G(s) = \frac{K_p}{s(1 + T_{p1}s)} \tag{13}$$

In which $Kp = -1.641 \cdot 10^{11}$, and $T_{p1} = 8.774 \cdot 10^{9}$. The model performance is given in Figure 7.

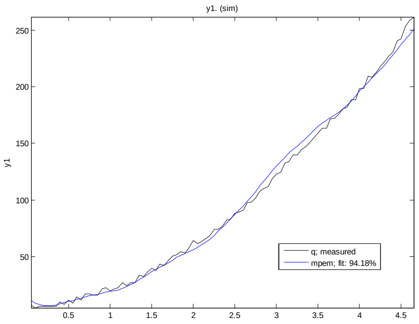


Figure 7. **Result of integer-order identification**

3. Fractional-order Controller Design

Due to the integrator within the process model, a PD controller is enough to control the process for an ideal system where there is no input disturbance. However, in practice, as with most industrial applications, a disturbance always exists (Figure 8). Hence, a proportional-integral-derivative PID controller is necessary in order to reject such input disturbance, otherwise the output of the system will have a steady-state error.

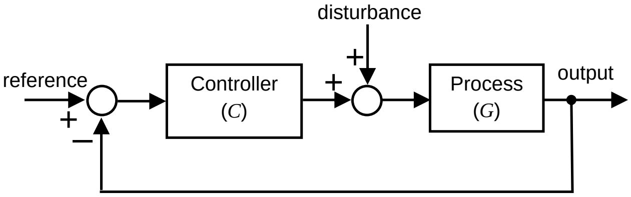


Figure 8. **Closed-loop structure of the system**

Based on the identified model from (13), two fractional-order $PI^{\lambda}D^{\mu}$ controllers are designed using: i) genetic algorithm and ii) auto-tuning method. In industrial applications, PID controllers are the most widely used feedback controller. Standard textbook representation of a PID controller is given by:

$$u(t) = K_p e(t) + K_i \int_0^t e(\tau)d\tau + K_d \frac{d}{dt} e(t) \tag{14}$$

and the continuous transfer function of the PID controller is obtained by means of the Laplace transformation as given by:

$$C(s) = K_p + \frac{K_i}{s} + K_d s. \tag{15}$$

Generalization of (15) leads to:

$$C(s) = K_p + \frac{K_i}{s^\lambda} + K_d s^\mu. \tag{16}$$

Comparing (15) with (16), it can be noticed that a fractional-order controller provides more flexibility in the controller design because it has five parameters to select in order to fulfill some desired specifications. However, this also implies that the tuning procedure will be more complex.

3.1. Tuning in the Time Domain

Many advanced optimization techniques like particle swarm optimization [23, 24] and genetic algorithm (GA) [25, 26] are applied to tune the $PI^\lambda D^\mu$ controller in the time domain. Here we will discuss the implementation of GA to tune the $PI^\lambda D^\mu$ controller.

The genetic algorithm is a stochastic search algorithm that mimics the mechanism of natural selection and natural genetics [27]. This heuristic procedure is routinely used to generate useful solutions for optimization and search problems. Genetic algorithms belong to the largest class of evolutionary algorithms, which generate solutions for optimization problems using techniques inspired by natural evolution, such as inheritance, mutation, selection, and crossover [27].

Unlike conventional search techniques, a GA starts with an initial set of random solutions, called a population, satisfying boundary and/or system constraints to the problem. Each individual in the population is called a chromosome, which represents a possible solution to the problem. Usually, a chromosome is a string of symbols, but not necessarily a binary bit string. The chromosomes evolve through successive iterations called generations. During the evolution from one generation to another generation, the chromosomes are evaluated by a function to obtain a fitness value. To create the next generation, new chromosomes, which are called offspring, are formed by i) either merging two chromosomes from the current generation using a crossover operator ii) either by modifying a chromosome using a mutation operator. The selection of parent chromosomes is based on the fitness values i.e. fitter chromosomes will have higher probability of being selected. As s result, a new generation is formed by ranking all the parent chromosomes and offspring. The fittest ones will be kept constant according to the population size. The following steps will guide the reader through a single iteration of a simple genetic algorithm [28]:

1. Evaluate the fitness of all chromosomes in the current population.
2. Select parent chromosomes from the population (with probability proportional to fitness).
3. Crossover and/or mutate parent chromosomes to form offspring.
4. Add offspring chromosomes to the next generation.

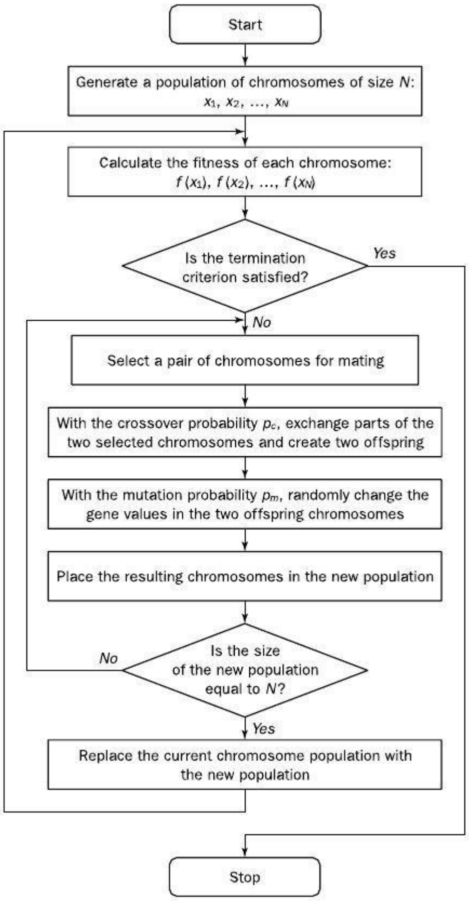

Start
Generate a population of chromosomes of size N: x1, x2, ..., xN
Calculate the fitness of each chromosome: f (x1), f (x2), ..., f (xN)
Is the termination criterion satisfied?
Yes
No
Select a pair of chromosomes for mating
With the crossover probability pc, exchange parts of the two selected chromosomes and create two offspring
With the mutation probability pm, randomly change the gene values in the two offspring chromosomes
Place the resulting chromosomes in the new population
No
Is the size of the new population equal to N?
Yes
Replace the current chromosome population with the new population
Stop

Figure 9. **The flowchart structure of a genetic algorithm**

5. Remove enough chromosomes from the next generation (with probability of being removed inversely proportional to fitness) to restore the number of population to N.

There are various time domain integral performance indices which can be used as fitness functions [29]. From control system designer's point-of-view, the Integral of square error (ISE) is the most popular. This performance index represents the H_2-norm of the closed loop system in the frequency domain. In this paper, the ISE is used to indicate the individual fitness value as:

$$Fitness = \int e^2(t)dt, \tag{17}$$

where e is the error between the system step response and the step reference. The corresponding GA parameters are given in Table 1.

Table 1. **GA parameters**

Number of Decision Variables	5 $(K_p, K_d, \mu, K_i, \lambda)$
Number of Individuals	50
Maximum Number of Generations	50
Number of Objectives	1
Selection Method	Tournament Selection
Crossover Rate	0.9
Mutation Rate	0.1

3.2. Tuning in the Frequency Domain

In the frequency domain, the $PI^\lambda D^\mu$ controller from (16) can be written as:

$$C(j\omega) = K_p + \frac{K_i}{\omega^\lambda \cos\frac{\lambda\pi}{2} + j\omega^\lambda \sin\frac{\lambda\pi}{2}} + K_d\left(\omega^\mu \cos\frac{\mu\pi}{2} + j\omega^\mu \sin\frac{\mu\pi}{2}\right) \tag{18}$$

and the system needs to fulfill specifications as [30]:

1. Phase margin φ_m specification:

$$\varphi_m = arg\left[C(j\omega_{cg})G(j\omega_{cg})\right] + \pi \tag{19}$$

2. Gain crossover frequency ω_{cg} specification:

$$|C(j\omega_{cg})G(j\omega_{cg})|_{\text{dB}} = 0 \tag{20}$$

3. Robustness to variations in the gain of the plant:

$$\left.\frac{d\left(arg\left(C(j\omega_{cg})G(j\omega_{cg})\right)\right)}{d\omega}\right|_{\omega=\omega_{cg}} = 0 \tag{21}$$

4. High-frequency noise rejection:

$$\begin{gathered}\left|T(j\omega) = \frac{C(j\omega)G(j\omega)}{1+C(j\omega)G(j\omega)}\right|_{\text{dB}} \le A\,, \\ \forall\omega \ge \omega_t \Rightarrow |T(j\omega_t)|_{\text{dB}} = A\end{gathered} \tag{22}$$

where $T(j\omega)$ is a sensitivity function and A the desired noise attenuation for frequencies $\omega \ge \omega_t$ rad/s.

5. Disturbance rejection:

$$\begin{gathered}\left|S(j\omega) = \frac{1}{1+C(j\omega)G(j\omega)}\right|_{\text{dB}} \le B\,, \\ \forall\omega \le \omega_s \Rightarrow |S(j\omega_s)|_{\text{dB}} = B\end{gathered} \tag{23}$$

with $S(j\omega)$ a sensitivity function and B the desired value of the sensitivity function for frequencies $\omega \le \omega_s$ rad/s.

6. Zero steady-state error. The closed loop system exhibits zero steady-state error if $\lambda > 0$ [30].

If for fractional-order controllers such as PI^λ or PD^μ, only three specifications must be met, the design problem of a $PI^\lambda D^\mu$ controller relies on solving a system with five nonlinear equations and five unknown parameters $(K_p, K_i, \lambda, K_d, \mu)$. This method implies a set of complex equations related to the design specifications and the obtained solution may not be the optimal one. Thus, for the tuning of the $PI^\lambda D^\mu$ controller we use the auto-tuning method proposed in [30], which provides an analytical solution.

Hence, the $PI^\lambda D^\mu$ controller can be written as a Lead-Lag Compensator:

$$C(s) = K_c \eta^\mu \left(\frac{\zeta s + 1}{s}\right)^\lambda \left(\frac{\vartheta s + 1}{\eta \vartheta s + 1}\right)^\mu, \tag{24}$$

which implies:

$$PD^\mu(s) = K_c \eta^\mu \left(\frac{\vartheta s + 1}{\eta \vartheta s + 1}\right)^\mu, \tag{25}$$

$$PI^\lambda(s) = \left(\frac{\zeta s + 1}{s}\right)^\lambda. \tag{26}$$

By analogy to standard PID form it follows: $K_p = k'(\zeta)^{-\lambda}$, $T_d = \vartheta(1-\eta)$, $T_i = \zeta$. The fractional-order PD^μ controller (25) corresponds to the lead part of the compensator, while the lag part can be identified as a fractional-order PI^λ controller (26). The controller design will be performed first for the $PI^\lambda + G$, then the $PD^\mu + (PI^\lambda + G)$. In the method proposed by [30], the integer controller is tuned in order to obtain a flat phase around the frequency point ω_{cg} and based on relay tests. In this way, the fractional-order PI^λ controller (26) is used to cancel the slope of the phase of the plant given by:

$$\upsilon = \frac{\phi_u - \phi_{n-1}}{\omega_u - \omega_{n-1}} \tag{27}$$

where ϕ_u is the plant phase corresponding to the frequency of interest $\omega_u = \omega_{cg}$ and ω_{n-1} is the frequency after $n-1$ experiments with the relay test and ϕ_{n-1} its corresponding plant phase.

The phase and magnitude of the plant in the frequency domain can be expressed as:

$$arg\left(G(j\omega_u)\right) = -\pi + \omega_u \delta \tag{28}$$

$$|G(j\omega_u)| = \frac{\pi a}{4d} \tag{29}$$

where d is the relay amplitude, a is the amplitude of the output signal and $G(j\omega_u)$ is the transfer function of the plant at the frequency ω_u. In (28), ω_u is the frequency of the output signal relative to the delay δ. In order to get a value of the frequency of the output signal approximately equal with a specific frequency ($\omega_u \simeq \omega_{cg}$), the corresponding value of δ is determined using an iterative method [30]:

$$\delta = \frac{\omega_u - \omega_{n-1}}{\omega_{n-1} - \omega_{n-2}} \left(\delta_{n-1} - \delta_{n-2}\right) + \delta_{n-1}, \tag{30}$$

where n is the current number of iteration. The corresponding frequency ω_u together with (28) are used in (27) to determine the slope of the phase of the plant. The phase of the fractional PI^λ controller is $\psi = \lambda\left(\arctan(\zeta\,\omega) - \frac{\pi}{2}\right)$. The derivative of the phase of the fractional-order PI^λ controller at the frequency point ω_{cg} $\left(i.e.\ \lambda\frac{\zeta}{1+(\zeta\,\omega_{cg})^2}\right)$ must be equal with $-\upsilon$ in order to cancel the slope of the phase curve of the plant. The derivative of $\lambda\frac{\zeta}{1+(\zeta\,\omega_{cg})^2}$ with respect to parameter ζ has a maximum at $\zeta = \frac{1}{\omega_{cg}}$. Choosing the zero frequency $\omega_{zero} = \frac{1}{\zeta} = \omega_{cg}$, the slope of the plant at the frequency ω_{cg} will be canceled with the maximum slope of the fractional-order controller.

Once ζ is available, the fractional order λ is given by:

$$\lambda = \frac{-\upsilon\left(1+(\zeta\omega_{cg})^2\right)}{\zeta}. \tag{31}$$

The PD^μ controller is designed in such a way that the open-loop system satisfies the phase margin φ_m and the gain crossover frequency ω_{cg} specifications. Now, if we define the new process as $F(s) = G(s)PI^\lambda(s)$, the open-loop transfer function of the new loop is $L(s) = F(s)PD^\mu(s)$.

In the complex plane, the next relation for the open-loop can be defined:

$$\begin{aligned} F(j\omega_{cg})k'\left(\frac{j\vartheta\omega_{cg}+1}{j\eta\vartheta\omega_{cg}+1}\right)^{\mu} &= e^{j(-\pi+\varphi_m)} \\ \Rightarrow \left(\frac{j\vartheta\omega_{cg}+1}{j\eta\vartheta\omega_{cg}+1}\right) = \frac{e^{j(-\pi+\varphi_m)}}{F(j\omega_{cg})k'} &= a_1 + jb_1 \\ \Rightarrow \left(\frac{j\vartheta\omega_{cg}+1}{j\eta\vartheta\omega_{cg}+1}\right) = (a_1+jb_1)^{\frac{1}{\mu}} &= a + jb \end{aligned} \tag{32}$$

where (a_1, b_1) is the "design point" [30] and $k' = K_c\eta^\mu$ is the value of the compensator gain and is set to 1 in this case .

It follows that the parameters η and ϑ can be computed using:

$$\eta = \frac{a-1}{a(a-1)+b^2},\ \vartheta = \frac{a(a-1)+b^2}{b\omega_{cg}}. \tag{33}$$

The fractional order μ is selected from the interval $[\mu_{min}, 1]$, where μ_{min} is obtained using an iterative method. Starting from a small value (e.g. $\mu_{min} = 0.03$), the fractional order μ_{min} is increased with a small step until the value of η computed with (33) is positive. The system robustness is higher when η is closer to zero, thus the optimal solution for fractional order is $\mu = \mu_{min}$ at the end of the iteration procedure. More details about the auto-tuning of a $PI^\lambda D^\mu$ controller can be found in [30].

4. Implementation details

4.1. Implementation of the controllers

One of the most common problems of fractional-order controllers is their implementation. In order to implement a fractional-order controller or to perform a simulation with a fractional-order controller, the fractional-order transfer functions are replaced by an integer-order transfer functions that have the same behavior as the fractional-order. In literature, different methods are used to find such approximations [31], but there is no criterion that can say which of them is the best. In this paper the Modified Oustaloup Filter was used to find the approximations of fractional-order controllers. The modified filter is given by [30]:

$$s^{\mu} \approx \left(\frac{d_1\omega_h}{d_2}\right)^{\mu}\left(\frac{d_1s^2+d_2\omega_h s}{d_1(1-\mu)s^2+d_2\omega_h s+d_1\mu}\right)\prod_{k=-N}^{N}\frac{s+\omega_b\omega_l^{(2k-1-\mu)/N}}{s+\omega_b\omega_l^{(2k-1+\mu)/N}} \tag{34}$$

where $\mu \in (0,1]$, d_1 and d_2 are same tuning parameters [30], N is the order of approximation and $\omega_l = \sqrt{\omega_h/\omega_b}$ with ω_b and ω_h the lower and upper limits of the frequency range. Equation (34) is used to approximate the derivative term and the integral term from (16). The numerical realization of $PI^{\lambda}D^{\mu}$ controllers uses the methods of the Tustin operator and the above continued fraction expansion.

4.2. Experimental results

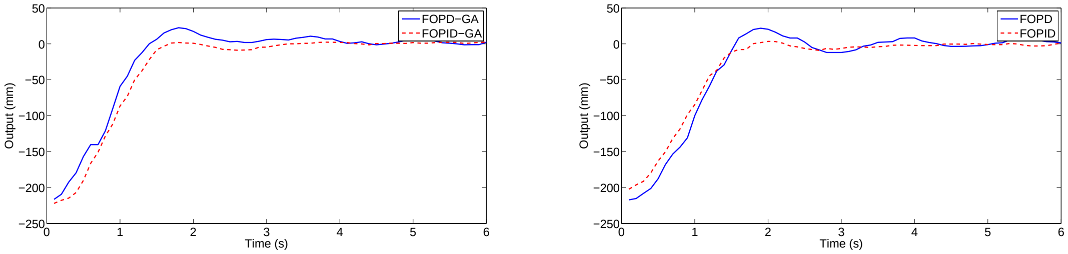
A Steward platform with 6 d.o.f. and a visual sensor mounted on the top of the table was considered for the real time experiments. The controllers obtained with the two tuning methods are implemented, tested and validated. Table 2 shows the tuning results obtained for PD^{μ} and $PI^{\lambda}D^{\mu}$ controllers with the two methods.

Table 2. **Controller tuning parameters**

parameters	PD^{μ}	PD^{μ}-GA	$PI^{\lambda}D^{\mu}$	$PI^{\lambda}D^{\mu}$-GA
K_p	-3.202	-2.609	-1.105	-1.008
K_d	-80.3	-136.21	-2.32	-1.991
K_i	0	0	0.6538	0.6718
λ	0	0	0.81	0.778
μ	0.728	0.782	0.412	0.407

Firstly, the controllers are compared to each other by their closed-loop step response, a move from 0 to 150 mm in the x-direction. As observed from Figure 10, when there is no disturbance the PD^{μ} controllers give good performance without steady-state error. Analyzing Figures 10(a)and 10(b) it can be noticed that the $PI^{\lambda}D^{\mu}$ controllers stabilize the system faster than the PD^{μ} controllers and have somewhat less overshoot. Both tuning methods give similar results.

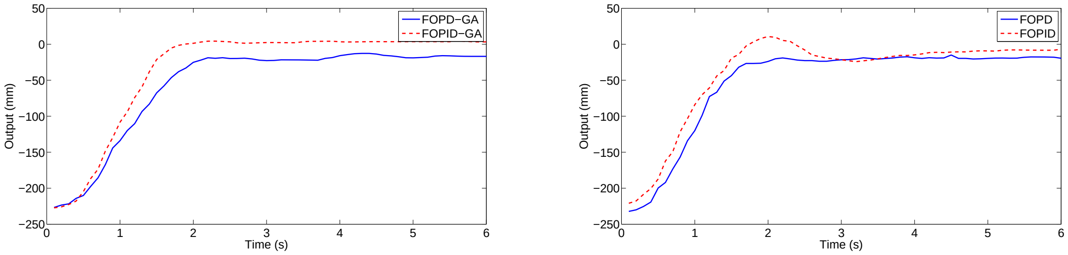
The second experiment uses the same platform, but the length ℓ of the shaft for two links is modified. Thus, an input disturbance is introduced (as in most of the real-life processes). The closed loop results are shown in Figure 11. It can be noticed that PD^{μ} controllers have steady-state error. Analyzing Figure 11(a) and 11(b) it can be observed that $PI^{\lambda}D^{\mu}$ controllers reject well the disturbance and have similar performance. One advantage of GA method over the numerical one, is that GA does not require numerical complexity.



(a)

(b)

Figure 10. The output of the system without disturbance for: (a) PD^{μ} and $PI^{\lambda}D^{\mu}$ controllers design with GA; (b) PD^{μ} and $PI^{\lambda}D^{\mu}$ controllers design with auto-tuning method



(a)

(b)

Figure 11. The output of the system with disturbance for: (a) PD^{μ} and $PI^{\lambda}D^{\mu}$ controllers design with GA; (b) PD^{μ} and $PI^{\lambda}D^{\mu}$ controllers design with auto-tuning method

5. Conclusions

In this paper, $PI^{\lambda}D^{\mu}$ controllers are implemented and tested on a ball and plate system. By changing the experimental conditions, i.e. change the length of the shaft for two links, we show the effectiveness of the $PI^{\lambda}D^{\mu}$ controllers to reject input disturbance. This result is very important since in the real-life applications, as with most of the mechatronic systems, a disturbance exists. Two tuning methods for designing the $PI^{\lambda}D^{\mu}$ controllers are presented. One method is based on the artificial intelligence to design the controller, while the other one tunes the controller in the frequency domain. The experimental results show that both tuning methods obtain good performances when dealing with mechatronic systems.

Acknowledgments

The authors would like to acknowledge ir. M. De Paepe for his technical assistance. Clara M. Ionescu is a post-doc fellow of the Research Foundation - Flanders (FWO).

References

[1] R.H. Bishop, The Mechatronics Handbook, (CRC Press, ISBN 0-8493-0066-5, 2002)

[2] M. Steinbuch, R.J.E. Merry, M.L.G. Boerlage, M.J.C. Ronde and M.J.G. van de Molengraft, Advanced Motion Control Design, In Levin, W.S. (Ed.), The Control Handbook, Control System Application, Second edition, (CRC Press, 2010)

[3] M. Steinbuch, G. Schootstra and O.H. Bosgra, Robust control of a compact disc mechanism, In Levine, W.S. (Ed.), The control handbook, Control System Application, (CRC Express, 2000).

[4] K.J. Astrom, and T. Hagglund, Advanced PID control, (ISA-The Instrumentation, Systems, and Automation Society, ISBN 1-5561-7942-1, 2006)

[5] B. Stuart, A history of control engineering: 1930-1955, (IET, ISBN 978-0-86341-299-8, 1993).

[6] A. Oustaloup, La Derivation Non Entiere: Theorie, Synthese et Applications, (Hermes, Paris, 1995, ISBN 2-8660-1456-1).

[7] I. Podlubny, Fraction order systems and $PI^{-\lambda}D^{\mu}$ controllers, IEEE Transaction of Automation Control, vol. 44, pp. 208-214, (1999).

[8] B.M. Vinagre, C.A. Monje, A.J. Calderon, and J.I. Suarez, Fractional PID controllers for industry application: a brief introduction, Journal of Vibration and Control, vol. 13, pp. 1419-1429, (2007).

[9] R.L. Bagley, and R.A. Calico, Fractional-order state equations for the control of viscoelastic damped structures, Journal of Guidance, Control and Dynamics, vol. 14(2), pp. 304-311, (1991).

[10] N.M.F. Ferreira and J.A.T. Machado, Fractional-order hybrid control of robotic manipulators, (Proceedings of the 11th International Conference on Advanced Robotics, Coimbra, Portugal, 2003) pp. 393-398.

[11] A. Kailil, N. Mrani, M.M. Touati, S. Choukri and N. Elalami, Low earth-orbit satellite attitude stabilization with fractional regulators, International Journal of Systems Science, vol. 35(10), pp. 559-568, (2004).

[12] F. Padula and A. Visioli, Tuning rules for optimal PID and fractional-order PID controllers, Journal of Process Control, vol. 21, pp. 69-81, (2011).

[13] M. Silva and T. Machado, Fractional order PD^{α} joint control of legged robots, Vibration and Control, vol. 12, 12, pp. 1483-1501, (2006).

[14] J. Villagra, B. Vinagre and I. Tejado, Data-driven fractional PID control: application to DC motors in flexible joints, (In IFAC Conference on Advanced in PID Control, 2012).

[15] H. Fan, Y. Sun and X. Zhang, Research on fractional order controller in servo press control system, (Proceedings of the 2007 IEEE International Conference on Mechatronics and Automation, Harbin, China, 2007) pp. 934-938.

[16] C., Ma, and Y., Hori, Fractional order control and its application of controller for robust two-inertia speed control, (Proceedings of the 4th International Power electronics and Motion Control Conference, vol. 3, Xian, China, 2004) pp. 1477-1482.

[17] N.L.S. Hashim, A. Yahya, T. Andromeda, M. Rafiq Abdul Kadir, N. Mahmud and S. Samion, Simulation of

PSO-PI Controrller of Dc Motor in Micro-EDM System for Biomedical Application, Procedia Engineering, vol. 41, pp. 805-811, (2012).

[18] C.M. Ionescu, J.T. Machado, and R. De Keyser, Frational-order impulse response of the respiratory system, Computer and Mathematics with Applications, vol. 62, pp. 845-854, (2011)

[19] Y. Kobayashi, T. Watanabe, T. Ando, M. Seki, and M.G. Fujie, Fractional impedance control for reproducing the material properties of muscle and its application in a body weight support system, (3rd IEEE RAS and EMBS International Conference on Biomedical Robotics and Biomechatronics, 2010) pp. 553-559.

[20] D. Steward, A platform with six degrees of freedom, (Proceedings of the Institution of Mechanical Engineering, vol. 180(1), June, 1965) pp. 371-386.

[21] W. Rekdalsbakken, The use of artificial intelligence in controlling a 6 DOF motion platform, (Proceedings of 21st European Conference on Modeling and Simulation, Prague, Czech Republic, 2007) pp. 471-476.

[22] L. Ljung, System identification: theory for the user, (Prentice-Hall, ISBN 0-1388-1640-9, 2007)

[23] J. Kennedy, R.C. Eberhart and Y. Shi, Swarm Intelligence, (Morgan Kaufmann, ISBN 1-55860-595-9, 2001).

[24] R.C. Eberhart and J. Kennedy, A new optimizer using particle swarm theory, (Proceedings of the 6th International Symposium on Micro Machine and Human Science, Nagoya, Japan, 4th-6th October, 1995), pp. 39-43.

[25] L.Y. Chang and H.,C. Chen, Tuning of fractional PID controllers using adaptive genetic algorithm for active magnetic bearing system, WSEAS Transactions on System, vol. 8, 99. 226-236, (2009).

[26] P. Rai, V. Shekher and O. Prakash, Determination of stabilizing parameter of fractional order PID controller using genetic algorithm, International Journal of Computational Engineering and Management, vol. 15(1), pp. 24-32, (2012).

[27] D. E. Goldberg, Genetic Algorithms in Search, Optimization, and Machine Learning. (Addison-Wesley Publishing Company, Inc., Reading, MA, January 1989).

[28] M. Gen, R.W. Cheng and L. Lin, Network Models and Optimization: Multiobjecive Genetic Algorithm Approach. (Springer, London, 2008).

[29] H. Unbehauen, Controller design in time-domain, Control Systems, Robotics and Automation, vol. 2, ISBN: 978-1-84826-591-2, (2009).

[30] C.A. Monje, Y.Q. Chen, B.M. Vinagre, D. Xue, and V. Feliu, Fraction-order System and Controls: Fundamentals and Applications, (Springer, ISBN 1-8499-6334-7, 2010).

[31] D.Y. Xue, C.N. Zhao, and Y.Q. Chen, A modified approximation method of fractional order system, (Proceedings of IEEE Conference on Mechatronics and Automationm, Luoyang, China, 2006) pp. 1043-1048.