Tuning and Digital Implementation of a Fractional-Order PD Controller for a Position Servo

Aleksei Tepljakov, Eduard Petlenkov, Juri Belikov, and Sergei Astapov

***Abstract*—Fractional-order calculus offers flexible computational possibilities that can be applied to control design thereby improving industrial control loop performance. However, before theoretical results can be carried over to an industrial setting it is important to study the effects of fractional-order control by means of laboratory experiments. In this paper, we study the practical aspects of tuning and implementing a fractional-order PD controller for position control of a laboratory modular servo system using FOMCON ("Fractional-order Modeling and Control") toolbox for MATLAB. We provide an overview of the tools used to model, analyze, and design the control system. The procedure of tuning and implementation of a suitable digital fractional-order controller is described. The results of the real-time experiments confirm the effectiveness of used methods.**

***Index Terms*—fractional calculus, position servo, pid controller, digital control, control optimization**

I. Introduction

FRACTIONAL-ORDER calculus is the generalization of conventional calculus, where the order α of integration or differentiation is not restricted to integer numbers [1]. This generalization offers interesting modeling possibilities. The number of applications where fractional-order calculus is used has been growing steadily in the last years [2].

Non-integer calculus is actively used in the field of control system design [3], [4]. Novel modeling opportunities allow to design efficient linear and nonlinear control strategies [2]. It is a well known fact, that PID-type controllers are ubiquitous in the industry [5], [6]. However, a conventional PID controller is inferior to a fractional-order PID controller due to extended tuning flexibility of the latter. This was experimentally confirmed in, e.g., [7], [8], [9].

Computer Aided Control System Design (CACSD) tools are readily available to assist engineers in the task of developing suitable controllers for particular plants. Notable examples include CRONE [10] and Ninteger [11] toolboxes for MATLAB/Simulink software. The FOMCON ("Fractional-order Modeling and Control") toolbox [12], [13] was recently developed to further expand the existing toolset as well as to provide new features.

In our previous work [14], [15], we focused primarily on the problem of extending an existing implementation technique [16], [17] to achieve a frequency bounded approximation of a fractional-order lead compensator similar to, e.g., Oustaloup's method [4], [10], [18], and provided an example where such a controller was obtained for a position servo model. In this work we complement these results by summarizing the design and implementation methods for an equivalent controller, namely a fractional-order PD$^\mu$ controller, using the FOMCON toolbox control design module. Thereby, in addition, we extend the results in [19], [20]. We also confirm the effectiveness of these methods experimentally using a modular servo system provided by INTECO [21].

The paper is organized as follows. In Section II the reader is introduced to fractional-order control, fractional-order system implementation method, and the corresponding software tools used to design and realize a digital fractional-order controller. The description of the controlled servo system is provided in Section III. In Section IV we provide the steps necessary to design and realize a suitable controller for the position servo system. An overview of the experimental platform and the results of real-time closed loop control are given in Section V. Some items for discussion are outlined in Section VI. Finally, conclusions are drawn in Section VII.

A. Tepljakov, E. Petlenkov, J. Belikov and S. Astapov are with the Department of Computer Control, Tallinn University of Technology, Ehitajate tee 5, 19086, Tallinn, Estonia (e-mails: {aleksei.tepljakov, eduard.petlenkov, juri.belikov, sergei.astapov}@ttu.ee).

II. Fractional-order Control

A. Introduction to Fractional-order Control

In the heart of fractional-order modeling lies the generalized non-integer order fundamental operator

$$
{}_a\mathscr{D}_t^\alpha = \begin{cases} \mathrm{d}^\alpha/\mathrm{d}t^\alpha & \alpha > 0, \\ 1 & \alpha = 0, \\ \int_a^t (\mathrm{d}\tau)^{-\alpha} & \alpha < 0, \end{cases} \tag{1}
$$

where a and t denote the limits of the operation. The case $\alpha \in \mathbb{Z}$ corresponds to conventional differentiation or integration. There exist several definitions of the generalized operator. Next, we provide the Grünwald-Letnikov definition [2], [3]:

$$
{}_a\mathscr{D}_t^\alpha f(t) = \lim_{h \to 0} \frac{1}{h^\alpha} \sum_{j=0}^{k} (-1)^j \binom{\alpha}{j} f(t - jh), \tag{2}
$$

where $a = 0$, $t = kh$, k is the number of computation steps and h is step size. We assume zero initial conditions and thus the Laplace transform of the fractional α-order derivative is

$$
\int_0^\infty \mathrm{e}^{-st} {}_0\mathscr{D}_t^\alpha f(t) \mathrm{d}t = s^\alpha F(s), \tag{3}
$$

where $\alpha \in \mathbb{R}^+$ and $s = \sigma + j\omega$ is the Laplace transform variable.

The parallel form of the fractional PI$^\lambda$D$^\mu$ controller is given in the following equation:

$$
C(s) = K_p + \frac{K_i}{s^\lambda} + K_d \cdot s^\mu. \tag{4}
$$

In the frequency domain this controller offers more tuning flexibility. In general, by varying the order γ of a fractional-order integrator (differentiator) a constant decrement (increment) in the slope of the magnitude curve that equals 20γ dB/dec can be achieved, as well as a constant delay in the phase plot $\pi\gamma/2$ rad, where the values depend on the sign of γ. The effects of control actions in the time domain corresponding to a fractional-order integrator and differentiator are illustrated in Fig. 1 and Fig. 2, respectively.

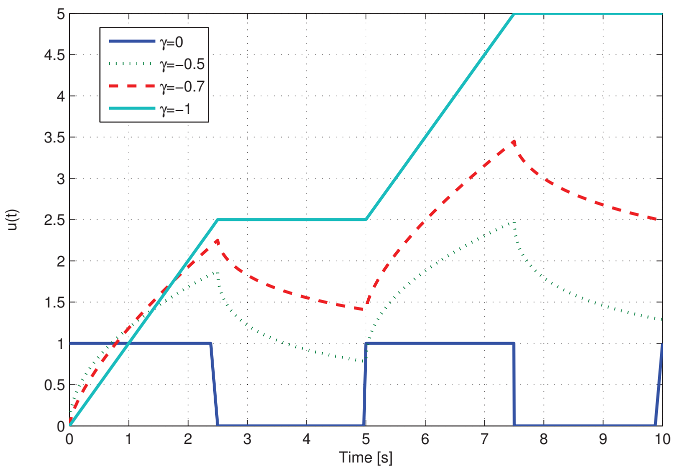

Fig. 1. Fractional integrator $s^{-\gamma}$ control actions

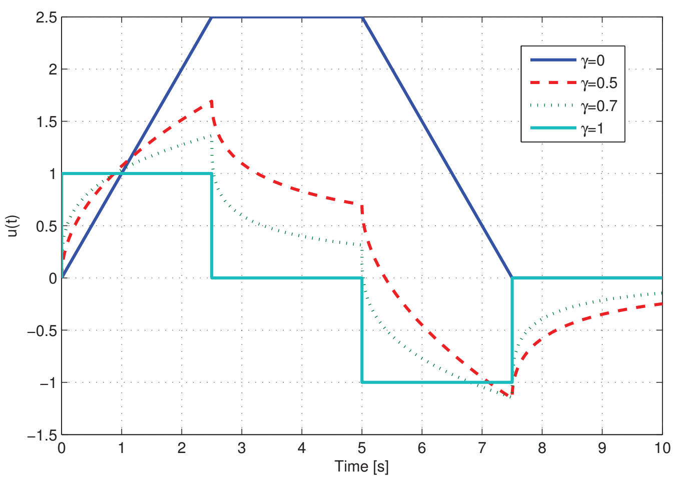

Fig. 2. Fractional differentiator s^{γ} control actions

Various fractional-order $\mathrm{PI}^{\lambda}\mathrm{D}^{\mu}$ controller tuning methods have been proposed to date [9], [22], [23], [24], [25]. One possible general approach is based on constrained optimization subject to particular control system design specifications and has been investigated in [26].

Additionally, consider the transfer function corresponding to a fractional-order lead (lag) compensator:

$$G(s) = K \cdot \left(\frac{bs+1}{as+1}\right)^{\alpha}, \tag{5}$$

where K is the static gain, $|\alpha| < 1$ is the non-integer power. Coefficients b and a are related to zero frequency $\omega_z = 1/b$ and pole frequency $\omega_h = 1/a$ for $\alpha > 0$. Generally, this transfer function corresponds to a frequency bounded non-integer differentiator (integrator) [15]. Therefore, the integral and differential components of the FOPID controller in (4) may be implemented using (5).

B. Fractional-order System Implementation

In this work we turn our attention to the Oustaloup approximation method which is frequently used for practical implementations of fractional-order systems and controllers [2], [3], [18]. A revised version of this method was proposed in [27]. We restrict our attention to the original approximation algorithm. In order to approximate a fractional differentiator of order α or a fractional integrator of order $(-\alpha)$ one can use the following set of equations:

$$s^{\alpha} \approx K \prod_{k=1}^{N} \frac{s+\omega'_k}{s+\omega_k}, \tag{6}$$

where

$$\omega'_k = \omega_b \cdot \omega_u^{(2k-1-\alpha)/N}, \tag{7}$$
$$\omega_k = \omega_b \cdot \omega_u^{(2k-1+\alpha)/N}, \tag{8}$$
$$K = \omega_h^{\alpha}, \quad \omega_u = \sqrt{\omega_h/\omega_b}, \tag{9}$$

and N is the order of approximation in the frequency range $(\omega_b; \omega_h)$. The order of the resulting approximation is $2N+1$. Taking a higher order N generally results in a more accurate approximation, though equations relating the parameters in (6)–(9) to N exist as well and may be found in [18].

A suitable discretization method can be used to obtain a discrete-time approximation from the continuous one in (6). One possible method would be that of zero-pole matching equivalents, where direct mapping of continuous zeros and poles to discrete-time is done by means of the relation

$$z = \mathrm{e}^{sT_s}, \tag{10}$$

where T_s is the desired sampling interval. The gain of the resulting discrete-time system $H(z)$ must be corrected by a proper factor. This implementation method has been successfully used in our previous work [28], [29]. We remark, that for the synthesis of continuous zeros and poles in (6) with the intent to obtain a discrete-time approximation the transitional frequency ω_h may be chosen such that

$$\omega_h \leqslant \frac{2}{T_s}. \tag{11}$$

After acquiring a set of discrete-time zeros and poles by means of (10), the fractional-order controller may be implemented in form of a IIR filter represented by a discrete-time transfer function $H(z^{-1})$. In general, one has two choices:

1) Implement each fractional-order component approximation of the controller in (4) separately as $H^{\lambda}(z^{-1})$ and $H^{\mu}(z^{-1})$; this method offers greater flexibility, since the components may be reused in the digital signal processing chain, but requires more memory and is generally more computationally expensive;
2) Compute a single LTI object approximating the whole controller; this method is suitable when there is a need

for a static description of a fractional-order controller, e.g., for a given control task.

In this particular work we choose the second option, that is we seek a description of the controller in the form

$$H(z^{-1}) = K\frac{b_0 + b_1 z^{-1} + b_2 z^{-2} + \cdots + b_m z^{-m}}{a_0 + a_1 z^{-1} + a_2 z^{-2} + \cdots + a_n z^{-n}}. \tag{12}$$

For practical reasons, the equivalent IIR filter should be comprised of second-order sections, since this allows to improve computational stability, especially when the target signal digital processing hardware has limited floating-point type resolution and operation support [28], [30]. Thus, the discrete-time controller must be transformed to yield

$$H(z^{-1}) = b_0 \prod_{k=1}^{N} \frac{1 + b_{1k} z^{-1} + b_{2k} z^{-2}}{1 + a_{1k} z^{-1} + a_{2k} z^{-2}}. \tag{13}$$

The form also easily lends itself to stability analysis regardless of the method used to generate the coefficients of the second-order sections. Here we assume that computational stability is guaranteed with a specified precision. Then, in order to determine whether a section is stable or not, we consider its discrete-time pole polynomial

$$p(z^{-1}) = 1 + d_1 z^{-1} + d_2 z^{-2}. \tag{14}$$

A single section in (13) is stable [31], if the following conditions, derived from the classical case of discrete-time system stability analysis, are met in the (d_1, d_2)-plane (see Fig. 3 for a visual reference):

$$|d_1| < 1 + d_2, \quad |d_2| < 1. \tag{15}$$

Note, that in case of the method in (6) for any non-integer order $\alpha \in \mathbb{R}$, $|\alpha| < 1$, the obtained approximation is always stable [18] and after the subsequent discretization procedure the rule (15) should also be satisfied for every section comprising the filter in (13).

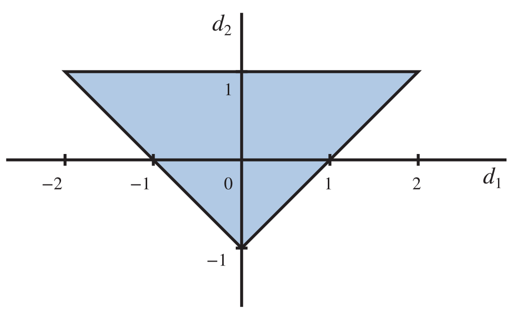

Fig. 3. The stability triangle

C. Fractional-order Controller Design Tools

The tools in the identification, control design, and implementation modules pertaining to the present work are briefly described next. The corresponding MATLAB calling sequence is provided.

- Fractional-order transfer function model identification tool, calling sequence: `fotfid`. The graphical user interface of the tool is presented in Fig. 4. Since fractional-order calculus is viewed as a generalization of the usual calculus operators in this context, the tool can also be used to determine the parameters of classical, integer-order systems. In particular, we are interested in identifying process models, a feature which the present tool fully supports.
- Fractional PID controller design front-end, calling sequence: `fpid`. A negative unity feedback connection is assumed, i.e.

$$G_c(s) = \frac{C(s)G(s)}{1 + C(s)G(s)},$$

where $C(s)$ is the fractional-order PID controller and $G(s)$ is the linear plant to be controlled. The tool has further links to the optimization and implementation tools.
- Optimization tool, calling sequence: `fpid_optim`. The tool has been updated substantially since [12], the new graphical front-end illustrating the added features is given in Fig. 5. A thorough description of the tool is provided in [26]. One of the most notable features is the possibility to use Simulink for simulation of nonlinear effects, such as actuator saturation, and/or nonlinear plants. In this regard, the nonlinear simulation option aims to fill the gap between theoretical controller design results and practical control applications.
- Implementation tools, calling sequence: `impid`, `d2sos()`. The former allows to choose suitable controller parameters for a discrete approximation, while the latter may be used to directly obtain C language style IIR filter second-order arrays from the approximated discrete controller.

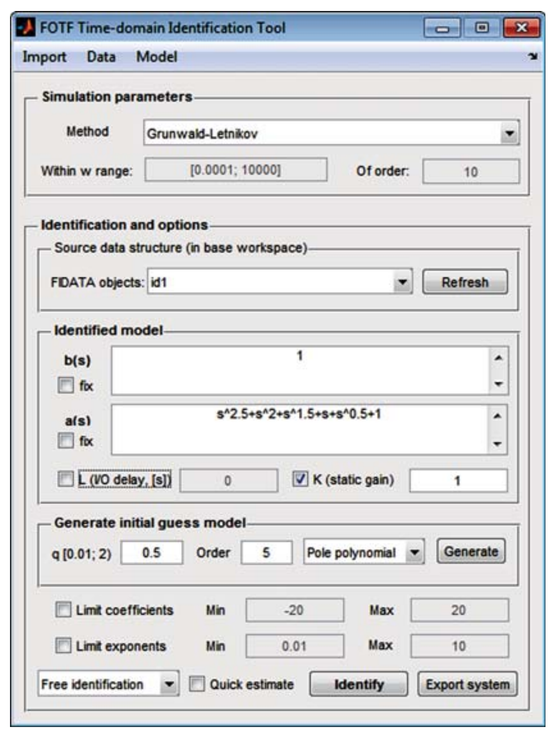

Fig. 4. Graphical user interface of the fractional-order transfer function model identification tool

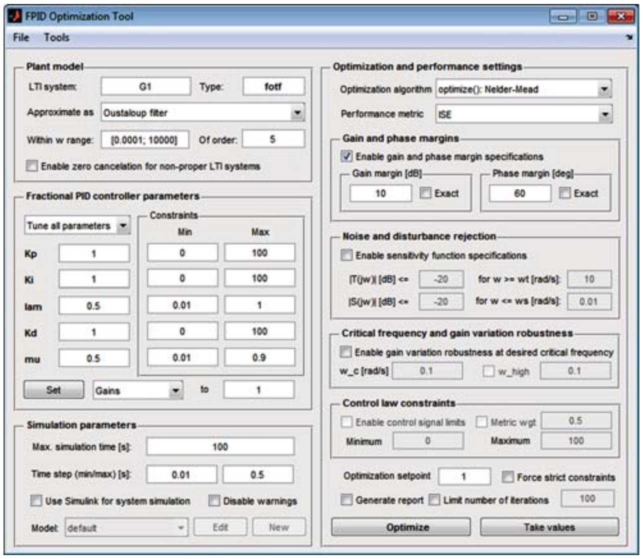

Fig. 5. Graphical user interface of the fractional-order PID controller optimization tool

III. Description of the Servo System

In this work we use the modular servo system provided by INTECO [21]. This laboratory system is reconfigurable and can be used for a variety of control experiments. We consider the particular configuration depicted in Fig. 6. The plant consists of the following modules: tachogenerator, which is used to measure the rotational speed of the DC motor; inertia load, backlash, incremental encoder, and gearbox with output disk. Data acquisition and real-time experiments are done using a specific PCI board, which connects to the servo system power interface and also collects data from the tachogenerator, incremental encoder and reference potentiometer (the latter is not shown in the figure).

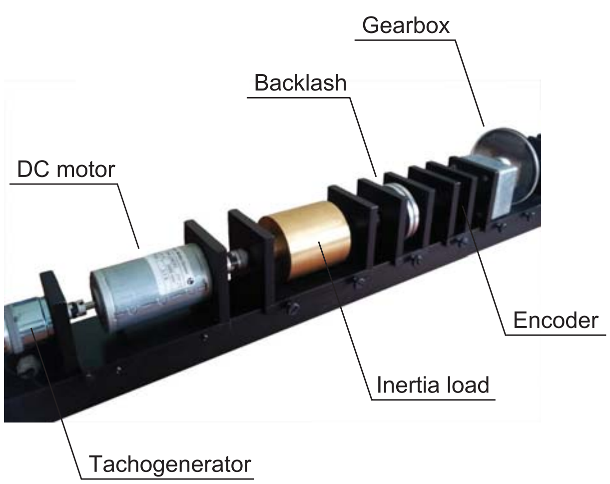

Fig. 6. Configuration of the laboratory servo system

The mathematical model of the servo system is based on that of the DC motor. The first-order inertial system, where static and dry kinetic friction and saturation are neglected, is described by the equation

$$T_s\dot{\omega}(t) = -\omega(t) + K_{sm}v(t), \tag{16}$$

where T_s is the time constant of the motor, $\omega(t)$ is the angular velocity of the rotor, K_{sm} is the motor gain and $v(t)$ is the input voltage. The input voltage in case of this laboratory plant is normalized and is such that

$$u(t) = v(t)/v_{max}, \quad |u(t)| \leqslant 1. \tag{17}$$

Also let $K_s = K_{sm} \cdot v_{max}$. Then, the corresponding velocity transfer function is given by

$$G_v(s) = \frac{K_s}{T_s s + 1}, \tag{18}$$

and the angle transfer function is obtained by adding an integrator

$$G_a(s) = \frac{K_s}{s(T_s s + 1)}. \tag{19}$$

The backlash, found in many real mechanical systems [32], introduces play into the system. In this plant the output dead-zone of the backlash is close to 2π. This nonlinearity has to be specifically considered when designing a suitable controller for the system—the procedure which we will investigate in the next section.

Finally, the apparent dead-zone of the motor under the voltage input has been identified as $u_d = (-0.05, 0.05)$.

IV. Tuning and Implementation of the Controller

In order to design the controller using the tools described in Section II we first need to obtain the model of the plant given by (19). This is possible by first identifying the model in (18) in the time domain from a step experiment, and then simply adding an integrator to arrive at the desired model. Using the time-domain identification tool, described in Section II, we obtain the following integer-order process model:

$$G(s) = \frac{192.1638}{s(1.001s + 1)}. \tag{20}$$

In what follows, this transfer function serves as the basis for controller design using the optimization tool.

For a model of the form (19) it is natural to assume that a lead compensator may be necessary for establishing the required closed-loop performance. Recall, that a fractional-order lead compensator corresponds, in principle, to a frequency-bounded approximation of a PD$^\mu$ controller, therefore our task is to establish an appropriate set of parameters (K_p, K_d, μ).

The generic parameters for the parallel form of this controller provided by INTECO are $K_p = 0.1, K_d = 0.01$. We shall use these parameters as the initial ones for the design of a suitable fractional-order PD controller, which we shall accomplish by means of constrained optimization. Following the controller design strategy applied in [26], [28], with design specifications of 1% set-point tracking accuracy and a minimum phase margin $\varphi_m = 60°$, we proceed to construct the nonlinear Simulink model for time-domain simulation of the control system. Additionally, input disturbance is considered, as well as the motor control dead-zone and the backlash component. The resulting model is presented in Fig. 7.

Other optimization parameters are set in the following way: performance metric is IAE (integral absolute error),

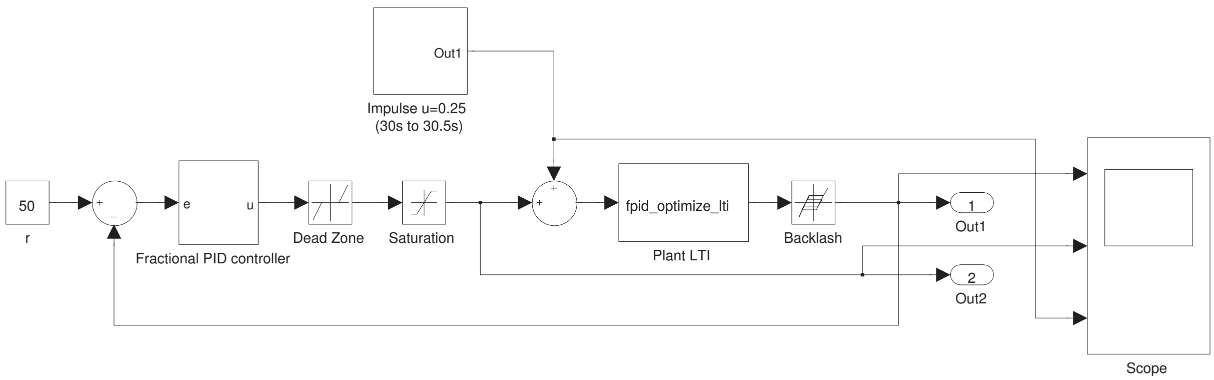

Fig. 7. Simulink model used for controller optimization with added nonlinearities and input disturbance

the linear model of the plant is used to compute the constraints in the frequency domain and the approximation is obtained by means of the Oustaloup filter with parameters $\omega \in [0.0001, 10000], N = 5$. Actuator saturation is considered and is such, that $u(t) \in [-1, 1]$. Simulation stop time is 60 seconds. Two optimization steps are considered:

- Fixing $\mu = 1$ we obtain optimized integer-order PD controller parameters K_p and K_d.
- Fixing the gains at the obtained values, we search for an optimal order μ.

This method minimizes the number of optimization variables thereby improving the optimization speed. The results of optimization are such, that after 100 iterations the gains of the PD controller have been found as $K_p = 0.055979$ and $K_d = 0.025189$. After fixing the gains and setting the initial value of μ to 0.5, the optimized PD$^\mu$ controller is obtained with $\mu = 0.88717$. Phase margin of the open loop control system is $\varphi_m = 65.3°$.

The fractional-order PD$^\mu$ controller has been found to exhibit superior performance than the initially obtained integer-order PD controller, especially in the presence of aforementioned nonlinearities. The comparison of simulated transient responses of the servo control system with initial generic integer-order PD controller and the optimized fractional PD$^\mu$ controller is given in Fig. 8.

The accuracy requirement as well as the phase margin specifications have been satisfied. In the frequency domain the approximation of the PD$^\mu$ controller corresponds to the fractional lead compensator in (5).

Finally, we can obtain a digital implementation of this controller. The target hardware is a simple 8 bit Atmel AVR ATmega8A based microcontroller prototype. The microcontroller connects to external 12 bit analog-to-digital and digital-to-analog converters by means of the I^{2}C interface. Basic signal conditioning circuits are employed.

Continuous Oustaloup filter approximation parameters are the same as were used during controller optimization. The `'matched'` method in 10 is used to obtain a discrete-time approximation of the controller with sampling interval $T_s = 0.01$. The `d2sos()` function is used to directly obtain the IIR filter second-order section coefficients, provided next:

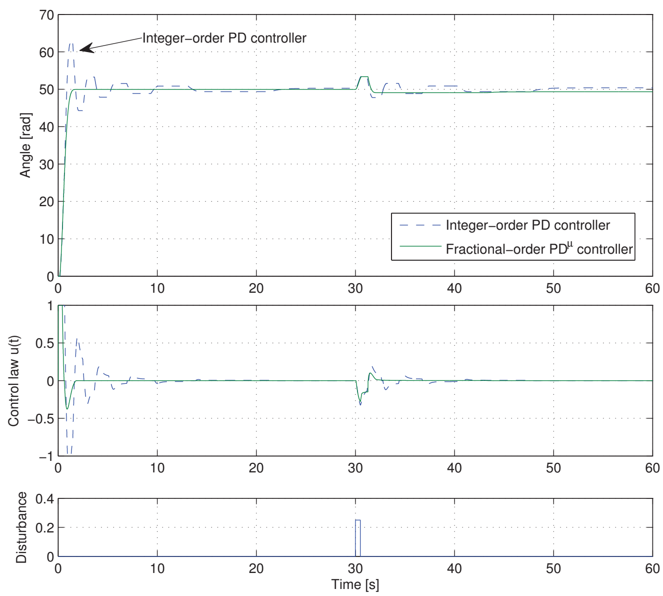

Fig. 8. Performance of the initial integer-order PD controller vs. the optimized PD$^\mu$ controller in the presence of an input disturbance

$$
\begin{aligned}
b \;=\; & \{+1.0000000000, -0.9647855878, +0.0000000000\}, \\
& \{+1.0000000000, -0.0209224276, +0.0000000000\}, \\
& \{+1.0000000000, -1.3493207288, +0.4180066451\}, \\
& \{+1.0000000000, -1.9807306143, +0.9807890156\}, \\
& \{+1.0000000000, -1.9991305017, +0.9991306026\}, \\
& \{+1.0000000000, -1.9999692428, +0.9999692429\},
\end{aligned}
$$

$$
\begin{aligned}
a \;=\; & \{+1.0000000000, -0.0000000000, +0.0000000000\}, \\
& \{+1.0000000000, -0.0409802515, +0.0000000016\}, \\
& \{+1.0000000000, -1.4434599048, +0.4912545169\}, \\
& \{+1.0000000000, -1.9752697983, +0.9753515564\}, \\
& \{+1.0000000000, -1.9991239831, +0.9991240851\}, \\
& \{+1.0000000000, -1.9999692318, +0.9999692319\},
\end{aligned}
$$

$$
b_0 = 1.5336084022.
$$

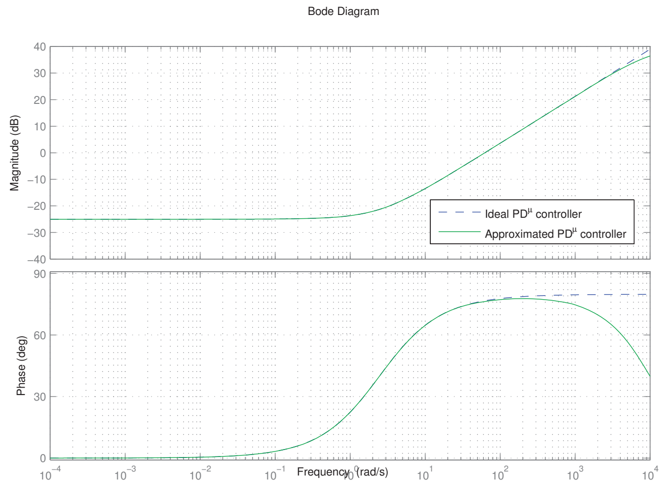

Fig. 9. Frequency domain characteristics of an ideal PD$^\mu$controller vs. its frequency-bounded approximation

These IIR filter coefficient arrays are hard-coded into the microcontroller memory. We remark, that the presented coefficient resolution will not be utilized in full by a single-precision floating number format used in DSP operations running on the microcontroller.

In the next section we describe the experimental platform and provide the results of real-time control experiments that verify the proposed implementation.

V. Experimental Results

A. Description of the Experimental Platform

In order to validate the performance of the digital controller, the configuration depicted in Fig. 10 is used. Apart from the servo system, we use an updated version of the serial communication based DAQ board, discussed in [14] in connection with controller prototyping. It offers two input and two output channels with 12 bit sample resolution and 2.5kSPS theoretical full-duplex real-time sampling rates on both channels with at most a single sample delay. Unfortunately, said performance will vary depending on the hardware configuration of the personal computer used.

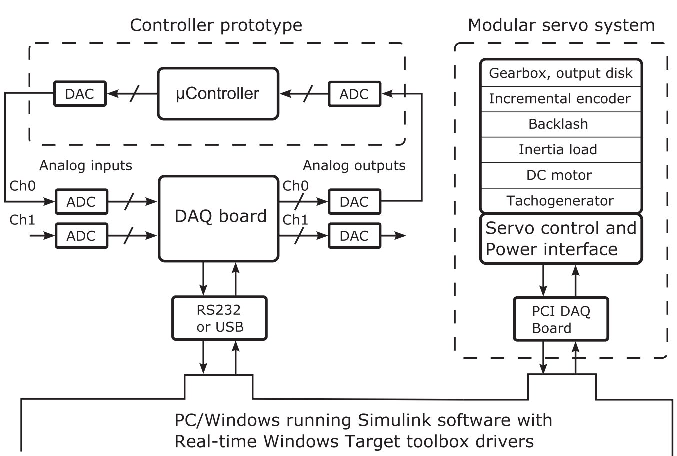

Fig. 10. Real-time closed loop control evaluation platform

For the experiment it is assumed, that the controller receives the error encoded in a voltage signal of amplitude $0 \ldots 5$V. Virtual ground with $+2.5$V reference is used to encode the negative error and control signal. The values are scaled accordingly. The voltage supply is reasonably well filtered.

The general Simulink diagram for experiments with the external controller is given in Fig. 11. In case of all experiments, the PC is running MATLAB/Simulink and specific Real-Time Windows Target drivers for the serial DAQ board and the modular servo system.

B. Real-time Control Results

Three experiments are considered:

- Evaluation of performance of the controller implemented in Simulink;
- Evaluation of performance of the external digital controller;
- Evaluation of external controller set-point tracking.

The first two experiments are grouped so that a comparison can be made. Additionally, a similar experiment is conducted with the initial integer-order PD controller for reference. The third experiment is done for the external controller. Set-point changing in this case is done by means of the potentiometer disk of the servo system.

The results of the first set of experiments are presented in Fig. 12. The control system responses obtained from using the fractional PD$^\mu$ controller implemented as a Simulink block and the external controller match up. A small discrepancy is caused by a voltage offset error. The control law exhibits limit cycles due to, one hand, quantization [33] caused by finite word length of the A/D and D/A converters, and on the other hand by measurement noise. Some noise is naturally present in the analog circuit. However, the amplitude of these limit cycles falls inside the dead zone of the control signal, so they do not have any major effect on the control system.

The result of manual set-point change experiment with the external controller is presented in Fig. 13. The experiment confirms the expected controller performance. We can conclude, that apart from small discrepancies, the hardware implementation of the PD$^\mu$ controller is working correctly within the desired performance specifications.

VI. Discussion

The methods of digital controller design and implementation provided in this paper were successfully verified by real-time control loop experiments. During this process some issues were identified.

- Limit cycles exist in the control law of the hardware controller due to noise and quantization effects. However, in an industrial application these will be minimized if a suitable data acquisition configuration tailored to the specific task is used.
- While the control system is capable of tracking the set angle with 1% accuracy, in a particular position tracking situation a more strict accuracy requirement may be demanded. Proper calibration must ensure the absence of offset errors.

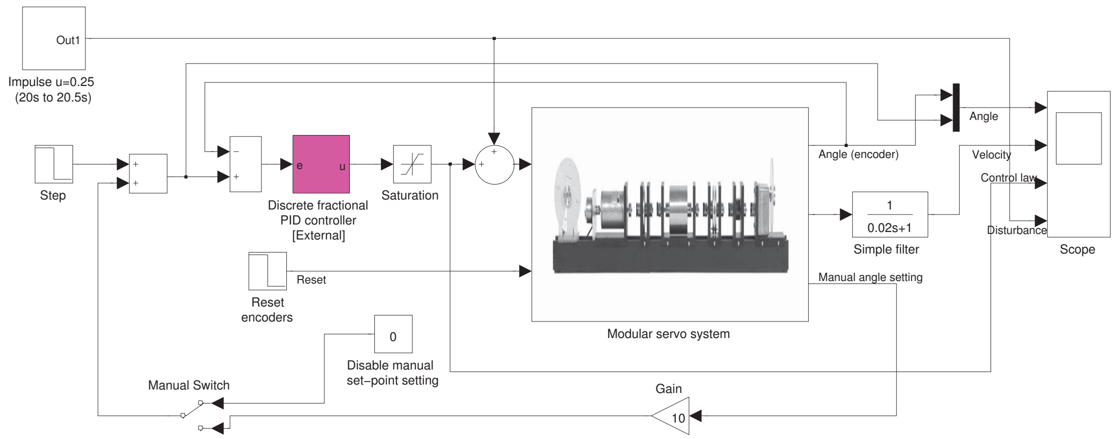

Fig. 11. Real-time closed loop control Simulink diagram

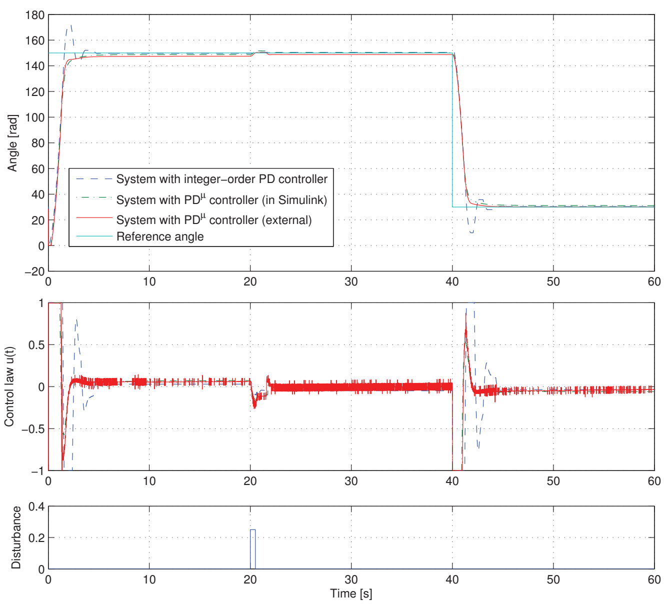

Fig. 12. Comparison of control systems with different controllers and configurations

VII. Conclusions

In this paper, we presented a method for design and implementation of a digital controller for a position servo using tools available in the FOMCON toolbox for MATLAB. The digital controller was implemented in form of a IIR filter running on an Atmel AVR microprocessor. The results of controller design were verified during real-time closed loop experiments. Some problems were identified and possible solutions provided. Future work should be devoted to further improving the controller design process, e.g. implementation of auto-tuning directly on the embedded device.

Acknowledgment

The work was supported by the European Union through the European Regional Development Fund, the target funding project SF0140113As08 of Estonian Ministry of Education and the ESF grant N8738, and the Estonian Doctoral School in Information and Communication Technology through the interdisciplinary project FOMCON.

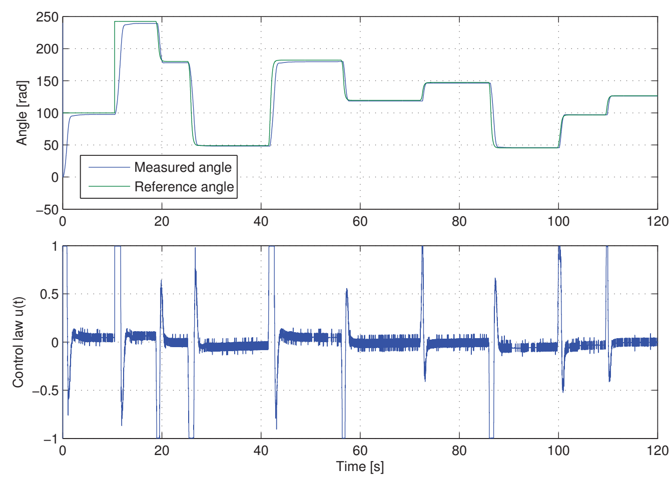

Fig. 13. External PD$^\mu$ controller set-point tracking evaluation

References

[1] K. Miller and B. Ross, *An introduction to the fractional calculus and fractional differential equations*. Wiley, 1993.

[2] C. A. Monje, Y. Chen, B. Vinagre, D. Xue, and V. Feliu, *Fractional-order Systems and Controls: Fundamentals and Applications*, ser. Advances in Industrial Control. Springer Verlag, 2010.

[3] Y. Q. Chen, I. Petráš, and D. Xue, "Fractional order control - a tutorial," in *Proc. ACC '09. American Control Conference*, 2009, pp. 1397–1411.

[4] A. Oustaloup, J. Sabatier, and P. Lanusse, "From fractal robustness to the CRONE control," *Fractional Calculus and Applied Analysis*, vol. 2, no. 1, pp. 1–30, 1999.

[5] K. Åström and T. Hägglund, *PID Controllers: Theory, Design, and Tuning*, 2nd ed. The Instrumentation, Systems, and Automation Society (ISA), 1995.

[6] ——, *Advanced PID control*. The Instrumentation, Systems, and Automation Society (ISA), 2006.

[7] M. Čech and M. Schlegel, "The fractional-order PID controller outperforms the classical one," in *Process control 2006*. Pardubice Technical University, 2006, pp. 1–6.

[8] D. Xue, C. Zhao, and Y. Chen, "Fractional order PID control of a DC-motor with elastic shaft: a case study," in *Proc. 2006 American Control Conference (ACC)*, 2006.

[9] C. Monje, B. Vinagre, V. Feliu, and Y. Chen, "Tuning and auto-tuning of fractional order controllers for industry applications," *Control Engineering Practice*, vol. 16, no. 7, pp. 798–812, 2008.

[10] A. Oustaloup, P. Melchior, P. Lanusse, O. Cois, and F. Dancla, "The CRONE toolbox for Matlab," in *Proc. IEEE Int. Symp. Computer-Aided Control System Design CACSD 2000*, 2000, pp. 190–195.

[11] D. Valério. (2005) Toolbox ninteger for MatLab, v. 2.3. [Online]. Available: http://web.ist.utl.pt/duarte.valerio/ninteger/ninteger.htm

[12] A. Tepljakov, E. Petlenkov, and J. Belikov, “FOMCON: a MATLAB toolbox for fractional-order system identification and control,” *International Journal of Microelectronics and Computer Science*, vol. 2, no. 2, pp. 51–62, 2011.
[13] ——. (2011) FOMCON toolbox. [Online]. Available: http://www.fomcon.net/
[14] ——, “Implementation and real-time simulation of a fractional-order controller using a MATLAB based prototyping platform,” in *Proc. 13th Biennial Baltic Electronics Conference*, 2012, pp. 145–148.
[15] ——, “Application of Newton’s method to analog and digital realization of fractional-order controllers,” *International Journal of Microelectronics and Computer Science*, vol. 2, no. 2, pp. 45–52, 2012.
[16] G. Carlson and C. Halijak, “Approximation of fractional capacitors $(1/s)^{1/n}$ by a regular Newton process,” *IEEE Transactions on Circuit Theory*, vol. 11, no. 2, pp. 210–213, 1964.
[17] B. M. Vinagre, I. Podlubny, A. Hernández, and V. Feliu, “Some approximations of fractional order operators used in control theory and applications,” *Fractional Calculus & Applied Analysis*, vol. 3, pp. 945–950, 2000.
[18] A. Oustaloup, F. Levron, B. Mathieu, and F. M. Nanot, “Frequency-band complex noninteger differentiator: characterization and synthesis,” *IEEE Transactions on Circuits and Systems—Part I: Fundamental Theory and Applications*, vol. 47, no. 1, pp. 25–39, 2000.
[19] I. Petráš, S. Grega, and L. Dorcak, “Digital fractional order controllers realized by PIC microprocessor: Experimental results,” in *Proc. of the ICCC’2003 conference*, High Tatras, Slovak Republic, 2003, pp. 873–876.
[20] I. Petráš, “Practical aspects of tuning and implementation of fractional-order controllers,” *ASME Conference Proceedings*, pp. 75–84, 2011.
[21] (2013) Official website of INTECO Sp. z o. o. [Last access time: 11.03.2013]. [Online]. Available: http://www.inteco.com.pl/
[22] D. Xue and Y. Chen, “A comparative introduction of four fractional order controllers,” in *Proceedings of the 4th World Congress on Intelligent Control and Automation*, 2002, pp. 3228–3235.
[23] C. Yeroglu, C. Onat, and N. Tan, “A new tuning method for $\mathrm{PI}^{\lambda}\mathrm{D}^{\mu}$ controller,” in *Proc. Int. Conf. Electrical and Electronics Engineering ELECO 2009*, vol. II, 2009, pp. 312–316.
[24] C. Zhao, D. Xue, and Y. Chen, “A fractional order PID tuning algorithm for a class of fractional order plants,” in *Proc. IEEE Int. Mechatronics and Automation Conf.*, vol. 1, 2005, pp. 216–221.
[25] C. A. Monje, B. M. Vinagre, A. J. Calderon, V. Feliu, and Y. Chen, “Auto-tuning of fractional lead-lag compensators,” in *Proceedings of the 16th IFAC World Congress*, 2005.
[26] A. Tepljakov, E. Petlenkov, and J. Belikov, “A flexible MATLAB tool for optimal fractional-order PID controller design subject to specifications,” in *Proceedings of the 31st Chinese Control Conference*, W. Li and Q. Zhao, Eds., Hefei, Anhui, China, 2012, pp. 4698–4703.
[27] D. Xue, Y. Chen, and D. P. Atherton, *Linear Feedback Control: Analysis and Design with MATLAB (Advances in Design and Control)*, 1st ed. Philadelphia, PA, USA: Society for Industrial and Applied Mathematics, 2008.
[28] A. Tepljakov, E. Petlenkov, J. Belikov, and M. Halás, “Design and implementation of fractional-order PID controllers for a fluid tank system,” in *Proc. 2013 American Control Conference (ACC)*, Washington DC, USA, June 2013, pp. 1780–1785.
[29] A. Tepljakov, E. Petlenkov, J. Belikov, and J. Finajev, “Fractional-order controller design and digital implementation using FOMCON toolbox for MATLAB,” in *Proc. of the 2013 IEEE Multi-Conference on Systems and Control conference*, 2013, pp. 340–345.
[30] A. V. Oppenheim, R. W. Schafer, and J. R. Buck, *Discrete-time signal processing (2nd ed.).* Upper Saddle River, NJ, USA: Prentice-Hall, Inc., 1999.
[31] S. Mitra, *Digital Signal Processing: A Computer-Based Approach*, ser. Connect, learn, succeed. McGraw-Hill Education, 2011.
[32] M. Nordin, “Controlling mechanical systems with backlash—a survey,” *Automatica*, vol. 38, no. 10, pp. 1633–1649, Oct. 2002.
[33] S. Kuo and B. Lee, *Real-time digital signal processing: implementations, applications, and experiments with the TMS320C55X.* Wiley, 2001.

Aleksei Tepljakov was born in 1987 in Tallinn. He received his B.Sc and M.Sc in Computer and Systems Engineering from Tallinn University of Technology in 2009 and 2011, respectively. He is currently a PhD student working at the Department of Computer Control at Tallinn University of Technology. His main research interests include fractional-order control of complex systems and fractional-order filter based analog and digital signal processing.

Eduard Petlenkov was born in 1979. He received his B.Sc, M.Sc and PhD degrees in computer and systems engineering from Tallinn University of Technology. He is an Associate Professor in the Department of Computer Control at Tallinn University of Technology. His main research interests lie in the domain of nonlinear control, system analysis and computational intelligence.

Juri Belikov was born in 1985. He received his B.Sc degree in mathematics from Tallinn University, and his M.Sc and PhD degrees in computer and systems engineering from Tallinn University of Technology. At present he holds the positions of researcher in the Institute of Cybernetics and Associate Professor in the Department at Computer Control of Tallinn University of Technology. His main research interests lie in the domain of nonlinear control theory.

Sergei Astapov was born in 1988. He received his M.Sc degree in the field of Computer System Engineering at the Tallinn University of Technology in 2011. He continues his education as a PhD student at the Department of Computer Control at the Tallinn University of Technology and is a member of the Department’s Research Laboratory for Proactive Technologies. His research interests include object tracking using wideband signal analysis, classification tasks and distributed computing in embedded multi-agent systems. His recent research concerns object localization and identification in open environments and acoustic signal based diagnostics of industrial machinery.