symmetry

MDPI

Article

Application of a Fractional Order PI Controller for a Speed Servo Drive Control

Pavol Bistak [1,*], Igor Bélai [1], Igor Bélai, Jr. [1], Damir Vrancic [2,3] and Mikulas Huba [1]

[1] Institute of Automotive Mechatronics, Faculty of Electrical Engineering and Information Technology, Slovak University of Technology in Bratislava, Ilkovičova 3, 841 04 Bratislava, Slovakia; igor.belai@stuba.sk (I.B.); igor_belai@stuba.sk (I.B.J.); mikulas.huba@stuba.sk (M.H.)
[2] Department of Systems and Control, Jozef Stefan Institute, Jamova cesta 39, 1000 Ljubljana, Slovenia; damir.vrancic@ijs.si
[3] Faculty of Industrial Engineering Novo mesto, Šegova ulica 112, 8000 Novo mesto, Slovenia
[*] Correspondence: pavol.bistak@stuba.sk

Abstract: This paper deals with the tuning of the parameters of a fractional-order PI controller for the speed control of an electric servo drive in which the torque is set by a torque generator. The controller parameters are tuned using the multiple dominant pole method (MDPM), while the fractional order integrator is approximated by the Oustaloup method. The input parameters required for tuning the controller using MDPM are calculated using the optimization algorithm presented in this paper. This algorithm selects the optimal parameters from a set of points in three-dimensional space, based on the symmetry around a central point. The controller tuning is performed for the normalized control loop model. The obtained optimized normalized fractional order PI controller can then be applied to a real servo drive with specific parameters. The proposed tuning was also verified experimentally, comparing the obtained closed-loop responses with those of the integer-order PI controller. Both simulation and experimental results showed a significant reduction in the integral of the absolute error at the disturbance step compared to a control loop using an integer-order PI controller. This results in a faster output response to load torque steps and a smaller control error in a real servo drive.

Keywords: control loop; fractional order controller; Oustaloup's approximation; servo drive; speed control

check for updates

Citation: Bistak, P.; Bélai, I.; Bélai, I., Jr.; Vrancic, D.; Huba, M. Application of a Fractional Order PI Controller for a Speed Servo Drive Control. *Symmetry* **2024**, *16*, 1543. https://doi.org/10.3390/sym16111543

Academic Editors: Aleksandar Djordjevic and Miladin Stefanović

Received: 31 August 2024
Revised: 7 November 2024
Accepted: 13 November 2024
Published: 18 November 2024

1. Introduction

Fractional order control is based on fractional calculus, a generalization of standard calculus that uses integral and derivative of fractional order. Fractional calculus defines an operator for the derivative and integral. This operator can be called the integrodifferentiator [1] and is given by the Formula (1), where a and t are the limits of the operation and μ is the order of the derivative or integration.

$$ {}_aD_t^{\mu} f(t) = \begin{cases} \dfrac{d^{\mu}}{dt^{\mu}} f(t) & \Re(\mu) > 0, \\ 1 & \Re(\mu) = 0, \\ \displaystyle\int_a^t f(\tau)(d\tau)^{-\mu} & \Re(\mu) < 0 \end{cases} \tag{1} $$

The Laplace transform $\mathcal{L}$ of the derivative/integral of a function $f(t)$ of fractional order μ is [2]:

$$ \mathcal{L}\left\{ {}_aD_t^{\mu} f(t) \right\} = s^{\mu} F(s) \tag{2} $$

where s is a Laplace operator and

$$ F(s) = \mathcal{L}\{f(t)\} \tag{3} $$

There are various approaches to the definition of the fractional order integrodifferentiation operator, such as the definitions by Grünwald-Letnikov, Riemann-Liouville, or Caputo [3]. Based on these definitions, numerical algorithms have been derived to compute the fractional order derivative/integral from a given number of input samples, e.g., as in [4].

Another way to implement a fractional order operator is to approximate it by integer and finite transfer functions, which are summarized in [2]. These approximations lead to transfer functions of integer order, where $N \in \mathbb{N}^{+}$ is the order of the approximation:

$$\begin{gathered} s^{\mu} \approx \frac{b_N s^N + b_{N-1} s^{N-1} + \cdots + b_0 s^0}{a_N s^N + a_{N-1} s^{N-1} + \cdots + a_0 s^0} = \frac{M_A(s)}{N_A(s)} \\ a_i \in \mathbb{R}, b_i \in \mathbb{R}, a_i \geq 0, b_i \geq 0 \quad \forall \quad i \in \langle 0, N \rangle \end{gathered} \tag{4}$$

where b_0 to b_N are the coefficients of the numerator and a_0 to a_N are the coefficients of the denominator of the transfer function. The numerator and denominator polynomials are denoted as $M_A(s)$ and $N_A(s)$.

Linear proportional-integral (PI) controllers are often used to control the torque and speed of electrical machines. These controllers are popular because of their simplicity and ease of implementation. Several tuning methods have been published that have proven themselves in practice.

In electric servo drives, control structures with a fractional-order PID controller (FOPID) [5] were tested, which has more degrees of freedom in terms of the number of adjustable parameters compared to the standard integer order PI or PID controllers. The generalized transfer function of the FOPID controller $G_c(s)$ is given in (5), where $U(s)$ is the controller output, $E(s)$ is the control error, k_p, k_i, and k_d are the gains of the proportional, integral, and derivative components, λ is the order of the integrator, and μ is the order of the derivative. Note that the parameters λ and μ can be real numbers.

$$G_c(s) = \frac{U(s)}{E(s)} = k_p + k_i \frac{1}{s^{\lambda}} + k_d s^{\mu}, \quad \lambda > 0, \mu > 0 \tag{5}$$

The following is a list of works in which a fractional order PI (FOPI) or PID (FOPID) controller is used to control an electric servo drive with different tuning procedures.

The speed control of a DC motor using a FOPID controller whose output is the reference rotor voltage is described in [6]. The tuning of the controller parameters is based on the specification of the phase margin of the open loop. The performance of the control loop was verified by simulation and without investigating the effect of the load torque on the motor speed.

A method for calculating the parameters of an FOPI speed controller for a servo drive with a torque generator is described in [7]. The phase margin method was used to tune the controller parameters, whereby the controlled system was replaced by a first-order system and an integrator. The characteristics of the speed control loop were tested experimentally on two servo drives. The first was a DC servo drive with rotor current control. The experimental servo drive with a vector-controlled permanent magnet synchronous motor (PMSM) was the second. The experiments showed a faster response to setpoint tracking compared to the servo drive with an integer order PI controller.

The FOPI controller was also used in [8] to control the speed of a PMSM with a torque generator, where integer-order PI controllers were used to control the flux and torque components of the stator current vector and the FOPI controller was used only for speed control (as in the experimental part of this paper). The order of the controller presented in [8] was time-dependent. This resulted in less overshoot during the setpoint change and a lower error in disturbance rejection than with the constant-order FOPI controller. The performance of the control loop was verified experimentally.

Fractional-order PID controllers were also used in motor speed and current vector torque cascade control loops in servo drive with PMSM [9]. The parameters of the controllers were calculated using particle swarm optimization. The simulation results of the speed control loop were compared with FOPID and PID controllers. A significantly smaller error was achieved with the FOPID controllers.

The use of FOPI controllers to control the speed and the stator current vector of the PMSM was presented in [10]. The control structure was supplemented by a disturbance observer. The parameters of the controllers were adjusted depending on the magnitude of the observed load torque. The performance of the closed-loop was verified by simulations. The results showed a faster response of the closed-loop and a reduction in oscillations due to the disturbance observer.

In [11], a fractional order PD position controller is used in a PMSM servo drive. The output of the controller is the setpoint for the torque component of the stator current vector. The steady state error caused by the disturbance is suppressed by a linear extended disturbance observer (LESO) of integer order. The parameters of the position controller were tuned using the phase margin method. In the paper, experimental results are compared with a servo drive using either an integer order PD controller or a fractional-order PD controller. The servo drive with an integer order position controller was characterized by overshoot of the tracking responses, while there was no overshoot with the fractional-order controller.

The application of the FOPI speed controller to a servo drive with a vector-controlled induction motor is described in [12]. The parameters of the FOPI controller were set to achieve a shorter settling time and less overshoot compared to a conventional PI speed controller, which was verified by simulation and on a test rig.

The application of the FOPID controller in a model for tracking control of a robot manipulator with two degrees of freedom was used in [13]. The controllers were tuned using the phase margin method. The authors compared fractional and integer order PID controllers. The system with the FOPID controller was found to be more robust to external disturbances, load variations, and noise in the feedback channel.

As can be seen from the above review, the use of a fractional order controller can improve the quality of control by reducing settling time and/or improving disturbance rejection (changes in load torque).

The aim of this work was to develop and verify experimentally a FOPI speed controller tuning method in a servo drive with a torque generator implemented. The method should be based on knowledge of the parameters of the torque generator and the mechanical subsystem of the servo drive, and its application should minimize the integral of the absolute error at a disturbance (load torque) step.

The novelty is that the FOPI speed controller is tuned to a specific implementation of a fractional order integrator. This means that the properties of a fractional order integrator, which is approximated by the Oustaloup method, are taken into account when tuning the controller. Besides the controller gains and the integrator order, the parameters of the controller tuning are also the lower and upper limits of the frequency band of the approximated fractional-order integrator. This approach was chosen because existing FOPI controller tuning methods only allow the controller gains and order of the integral component to be computed by assuming the implementation of an ideal fractional-order integrator. When the fractional-order integrator is approximated by a rational transfer function using the Oustaloup method, its frequency response matches that of the ideal integrator only in a certain frequency band. The lower limit of this band is usually close to zero. However, increasing the lower frequency of an approximated fractional-order integrator can improve the output response at the disturbance step. Our approach outperforms existing FOPID methods, especially in disturbance step responses. In addition, the tuning of the controller parameters is performed for a normalized process (gain and a time delay equal to one). The gain and other parameters of the FOPI controller, including the frequency band of the fractional-order approximated integrator, can then be recalculated during implementation based on the actual gain and time delay of the controlled process, which is a servo drive with an implemented torque generator. The actual parameters of the FOPI controller are calculated by a combination of analytical and optimization methods developed by the authors of the paper.

In this paper, it is assumed that an electric servo drive with an implemented torque generator is the controlled system. It can be represented by the integrator plus dead time (IPDT) transfer function. This way of representing an electromechanical system of a drive is given, for example, in [14]. The transfer functions of the process (system) are then

$$\begin{aligned} S_c(s) &= \frac{\omega(s)}{M_m^*(s)} = e^{-T_d s}\frac{K_s}{s} \\ S_d(s) &= \frac{\omega(s)}{M_L(s)} = \frac{K_s}{s} \end{aligned} \tag{6}$$

where the rotor angular velocity $\omega(s)$ is the output of the system, T_d is the transport delay (dead time), and K_s is the gain of the system. The reference motor torque $M_m^*(s)$ is the control variable and the load torque $M_L(s)$ is the disturbance variable. The transport delay T_d is determined by the transport delay of the torque generator T_{GM} and the length of the sampling period of the speed controller T_s in the discrete controller implementation as follows:

$$T_d = T_{GM} + T_s/2 \tag{7}$$

The torque generator is implemented in the electrical inverter with a special motor control algorithm. The actual control system, which is presented in Section 3.2, uses vector control [15] to control the motor torque.

The system gain K_s is the inverse value of the moment of inertia J, but for generalization, the variable K_s is used to denote the gain of the system:

$$K_s = J^{-1} \tag{8}$$

The structure of the speed control loop with a fractional-order PI controller is shown in Figure 1, where ω^* is the reference speed, K_p is the gain of the proportional term of the controller, K_i is the inverse of the integration time constant, and $F_I(s)$ is the reference speed filter. The filter is used to suppress the overshoot of the actual motor speed to the setpoint change ω^*. The overshoot is caused by zeros in the closed loop transfer function.

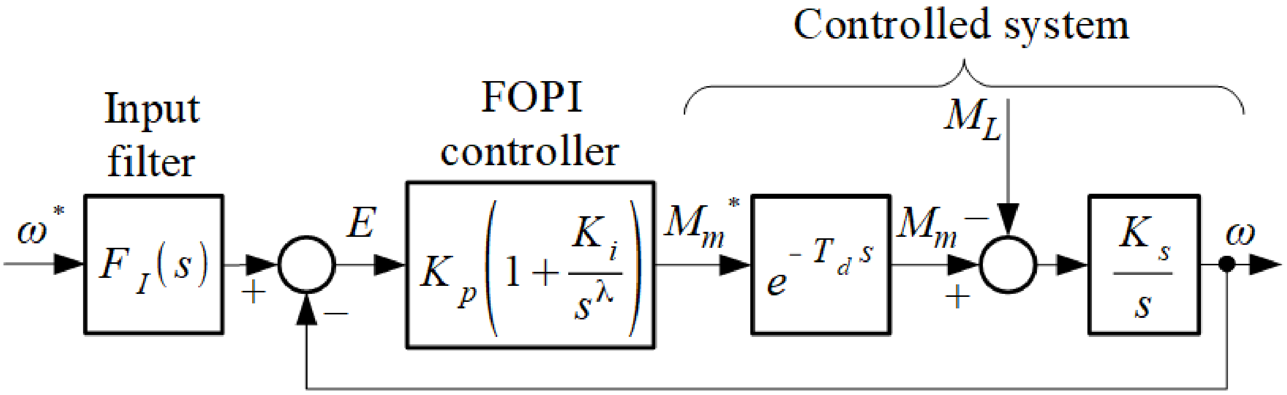

Figure 1. Speed control loop structure with fractional order PI (FOPI) controller.

If the fractional order integrator $1/s^\lambda$ is approximated by a transfer function of integer order (4), where $\mu = -\lambda \wedge F_I(s) = 1$, then the error transfer functions are:

$$\begin{aligned} G_{e,A}(s) &= \frac{E(s)}{\omega^*(s)} = \frac{se^{T_d s}N_A(s)}{se^{T_d s}N_A(s) + K_p K_i(N_A(s) + K_i M_A(s))} \\ G_{eL,A}(s) &= \frac{E(s)}{M_L(s)} = \frac{K_s e^{T_d s}N_A(s)}{se^{T_d s}N_A(s) + K_p K_s(N_A(s) + K_i M_A(s))} \end{aligned} \tag{9}$$

The steady-state errors for the unit reference speed and the load torque are calculated as follows

$$E_{\infty,A} = \lim_{s\to 0} G_{e,A}(s) = 0$$
$$E_{\infty L,A} = \lim_{s\to 0} G_{eL,A}(s) = \frac{K_s a_0}{K_p K_s (a_0 + K_i b_0)} \tag{10}$$

From the second formula in (10), it can be seen that if we want to eliminate the steady-state error ($E_{\infty L,A} = 0$), the lowest order coefficients of the numerator and denominator polynomials of the transfer function of the approximated integrator must be as follows:

$$a_0 = 0 \wedge b_0 > 0 \tag{11}$$

Expression (11) is fulfilled if the transfer function of the fractional-order integrator has the form of the product of the transfer functions of the first-order integrator and the fractional-order integrodifferentiator:

$$\frac{1}{s^{\lambda}} = \frac{1}{s} s^{\kappa}, \quad \lambda > 0, \quad \kappa = 1 - \lambda \tag{12}$$

The paper is organized as follows. First, the transfer functions of the controlled system and the structure of the control loop are presented above in the paper. Section 2.1 gives formulas to calculate the parameters of a fractional order integrator approximated by a rational transfer function. Section 2.2 then normalizes the process transfer function in the amplitude and time domains to simplify the derivation of the controller parameters (the actual controller gains can then be calculated by simple conversion according to the actual process gain and time delay). The tuning method for the normalized FOPI controller is described and presented in Section 2.3. The method for tuning an integer-order PI controller is given in Section 2.4. Section 2.5 describes the experimental workstation. The optimized values of the normalized parameters of the FOPI controller are presented in Section 3.1. A description of the experiments and experimental results can be found in Section 3.2. A discussion of the calculated parameters of the normalized control loop, a comparison of the experimental results with the expected properties of the control loop, and also a comparison of the properties of the speed loop tuned by the presented method with the properties of the loop tuned by the methods presented in the cited references are included in Section 4.

2. Materials and Methods

This section contains the following topics. 1. Transfer function of an approximated fractional-order integrator and transfer functions of a control loop. 2. The structure and transfer functions of the normalized control loop. 3. The FOPI controller tuning method. 4. The PI controller tuning. 5. Description of the experimental workstation.

2.1. Approximation of a Fractional Order Integrator by Oustaloup's Method

A method for approximating a fractional order operator by a transfer function of integer order was introduced by A. Oustaloup and published in [16]. This method was used for the implementation of the integral component of the PI controller in this work.

The transfer function $\widehat{G}_{o}^{\kappa}(s)$, which has an integer order N and is derived using Oustaloup's method, has the same frequency response in the specified frequency band ω_b to ω_h as the integrodifferentiator of fractional order s^{κ}. The higher the order N, the less wavy the frequency response of the transfer function $\widehat{G}_{o}^{\kappa}(s)$ and the smaller the difference to the frequency response of an ideal fractional order integrodifferentiator [17].

Various forms of the transfer function $\widehat{G}_o^{\kappa}(s)$ based on the work of [16] are described in the literature [1]. In this paper, the following form is assumed:

$$\begin{gathered} s^{\kappa} \approx \widehat{G}_o^{\kappa}(s) = \omega_h^{\kappa} \prod_{j=1}^{N} \frac{s + \gamma_j^{'}}{s + \gamma_j} \; , \quad \kappa \in \mathbb{R}, N \in \mathbb{Z}, N \geq 1 \\ \gamma_j^{'} = \omega_b \left(\frac{\omega_h}{\omega_b} \right)^{\frac{2j-1-\kappa}{2N}} , \; \gamma_j = \omega_b \left(\frac{\omega_h}{\omega_b} \right)^{\frac{2j-1+\kappa}{2N}} \end{gathered} \tag{13}$$

For a transfer function of an integer order that approximates the integrator (12) using the Oustaloup method (13), the following equations apply:

$$\begin{gathered} \frac{1}{s^{\lambda}} = \frac{1}{s} s^{1-\lambda} \approx \frac{M_I(s)}{N_I(s)} \; , \quad \lambda \in \mathbb{R} \, , \lambda > 0 \\ M_I(s) = K_o \prod_{j=1}^{N} s + \omega_j^{'} \; , \quad N_I(s) = s \prod_{j=1}^{N} s + \omega_j \\ K_o = \omega_h^{1-\lambda}, \; \omega_j^{'} = \omega_b \left(\frac{\omega_h}{\omega_b} \right)^{\frac{2j-2+\lambda}{2N}} , \; \omega_j = \omega_b \left(\frac{\omega_h}{\omega_b} \right)^{\frac{2j-\lambda}{2N}} \end{gathered} \tag{14}$$

After replacing the fractional order integrator $s^{-\lambda}$ with a rational transfer function according to (14), the transfer function of the PI controller becomes

$$G_c(s) = \frac{M_m^*(s)}{E(s)} = \frac{K_p}{\prod\limits_{j=1}^{N} s + \omega_j} \frac{s \prod\limits_{j=1}^{N} (s + \omega_j) + K_i K_o \prod\limits_{j=1}^{N} (s + \omega_j^{'})}{s} \tag{15}$$

which is the transfer function of a higher-order PID controller with a first-order integral term, derivative components of order 1 to N, and a filter with real poles $-\omega_1$ to $-\omega_N$. The closed-loop transfer functions with an integrator approximated by (14) when $F_I(s) = 1$ are

$$G_r(s) = \frac{\omega(s)}{\omega^*(s)} = \frac{K_p K_s N_I(s) + K_p K_i K_s M_I(s)}{s e^{T_d s} N_I(s) + K_p K_s N_I(s) + K_p K_i K_s M_I(s)} = \frac{M_{Or}(s)}{N_O(s)} \tag{16}$$

$$G_d(s) = \frac{\omega(s)}{M_L(s)} = -\frac{K_s e^{T_d s} N_I(s)}{s e^{T_d s} N_I(s) + K_p K_s N_I(s) + K_p K_i K_s M_I(s)} = -\frac{M_{Od}(s)}{N_O(s)} \tag{17}$$

Since the transfer function $G_r(s)$ in (16) contains zeros, the reference speed steps can lead to overshoots. To suppress the effect of the zeros and eliminate the overshoot, it is necessary to filter the reference speed signal (i.e., the setpoint signal) with $F_I(s)$. This allows a real pole $-s_0$ in the transfer function $G_r(s)$ to be compensated. The transfer function of the filter $F_I(s)$, whose order is $N + 1$, is as follows:

$$F_I(s) = \frac{s_0^{-1} s + 1}{N_I(s) + K_i M_I(s)} K_i K_o \prod_{j=1}^{N} \omega_j^{'} \tag{18}$$

If the speed reference signal ω^* is filtered with $F_I(s)$ (18), then the integral of the error (IE) for the setpoint step is

$$IE_r = \frac{\prod\limits_{j=1}^{N} \omega_j + K_i K_o \sum\limits_{i=1}^{N} \left(\frac{1}{\omega_i^{'}} \prod\limits_{j=1}^{N} \omega_j^{'} \right)}{K_i K_o \prod\limits_{j=1}^{N} \omega_j^{'}} - \frac{1}{s_0} \tag{19}$$

The IE for the disturbance step is as follows:

$$IE_d = \frac{\omega_b^{\lambda-1}}{K_p K_i} \tag{20}$$

2.2. Normalizing the Control Loop

By normalizing the control loop, the transport delay T_d can be replaced by a unit delay and the system gain K_s by a unit gain. By applying the following substitution:

$$\xi = T_d s \tag{21}$$

the control loop can be represented by Figure 2, with the overlines indicating the normalized parameters.

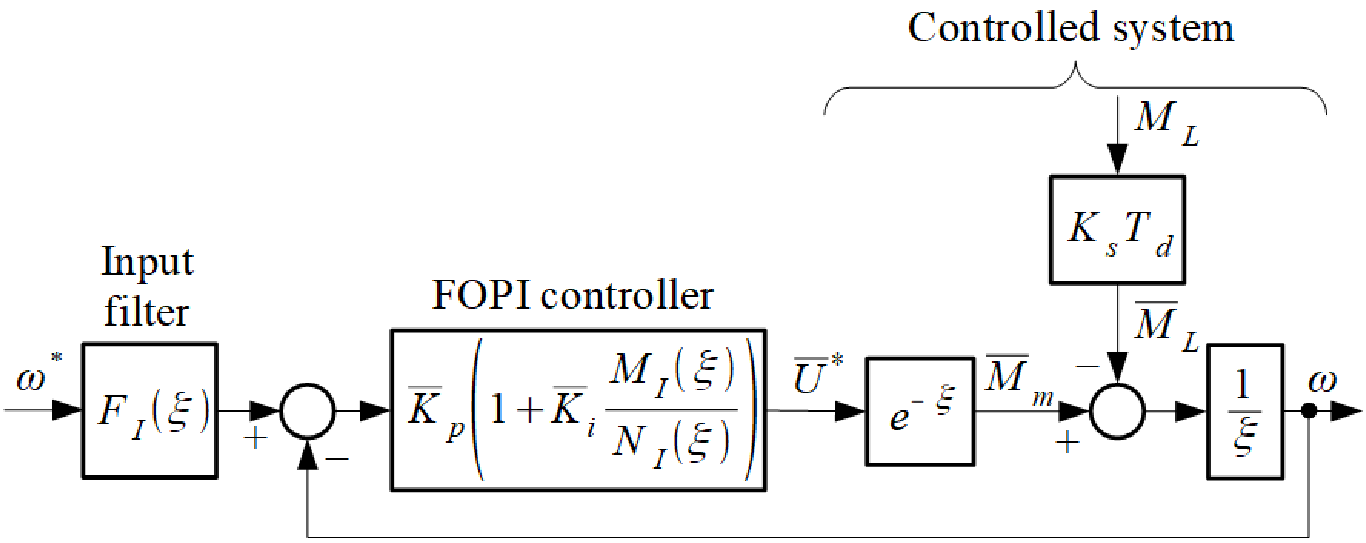

Figure 2. Structure of a normalized control loop with a fractional order PI controller with an integral component approximated by Oustaloup's method.

The formulas for calculating the normalized parameters of the controller and the integrator, which are approximated by Oustaloup's method (14), are as follows

$$\begin{gathered}
\overline{K}_p = K_p K_s T_d\,, \quad \overline{K}_i = K_i T_d^{\lambda}\,, \quad \overline{\omega}_b = \omega_b T_d\,, \quad \overline{\omega}_h = \omega_h T_d \\
\overline{\omega}_j = \overline{\omega}_b \left(\frac{\overline{\omega}_h}{\overline{\omega}_b}\right)^{\frac{2j-\lambda}{2N}} = \omega_j T_d\,, \quad \overline{\omega}_j' = \overline{\omega}_b \left(\frac{\overline{\omega}_h}{\overline{\omega}_b}\right)^{\frac{2j-2+\lambda}{2N}} = \omega_j' T_d \\
\overline{K}_o = \overline{\omega}_h^{1-\lambda} = K_o T_d^{1-\lambda}\,, \quad M_I(\xi) = \overline{K}_o \prod_{j=1}^{N} \xi + \overline{\omega}_j'\,, \quad N_I(\xi) = \xi \prod_{j=1}^{N} \xi + \overline{\omega}_j
\end{gathered} \tag{22}$$

The transfer functions of the normalized control loop without filtering the reference speed are:

$$G_r(\xi) = \frac{\omega(\xi)}{\omega^*(\xi)} = \frac{\overline{K}_p N_I(\xi) + \overline{K}_p \overline{K}_i M_I(\xi)}{\xi e^{\xi} N_I(\xi) + \overline{K}_p N_I(\xi) + \overline{K}_p \overline{K}_i M_I(\xi)} = \frac{M_{Or}(\xi)}{N_O(\xi)} \tag{23}$$

$$G_d(\xi) = \frac{\omega(\xi)}{\overline{M}_L(\xi)} = -\frac{e^{\xi} N_I(\xi)}{\xi e^{\xi} N_I(\xi) + \overline{K}_p N_I(\xi) + \overline{K}_p \overline{K}_i M_I(\xi)} = -\frac{M_{Od}(\xi)}{N_O(\xi)} \tag{24}$$

where

$$\begin{aligned} M_{Or}(\xi) &= \overline{K}_p N_I(\xi) + \overline{K}_p \overline{K}_i M_I(\xi) \\ M_{Od}(\xi) &= e^{\xi} N_I(\xi) \\ N_O(\xi) &= \xi e^{\xi} N_I(\xi) + \overline{K}_p N_I(\xi) + \overline{K}_p \overline{K}_i M_I(\xi) \end{aligned} \tag{25}$$

Similar to the structure shown in Figure 1, it is possible to filter the reference speed signal using a filter $F_I(\xi)$ that compensates the zeros of the transfer function $G_r(\xi)$ and one of its real poles $-\xi_0 = -T_d s_0$:

$$F_I(\xi) = \frac{\xi_0^{-1}\xi + 1}{N_I(\xi) + \overline{K}_i M_I(\xi)} \overline{K}_i \overline{K}_o \prod_{j=1}^{N} \overline{\omega}_j' \tag{26}$$

If the speed reference signal is filtered by $F_I(\xi)$, then the normalized integral of the error $\overline{IE}_r$ for the step setpoint signal is

$$\overline{IE}_r = \frac{\prod\limits_{j=1}^{N} \overline{\omega}_j + \overline{K}_i \overline{K}_o \sum\limits_{i=1}^{N} \left(\frac{1}{\overline{\omega}_i'} \prod\limits_{j=1}^{N} \overline{\omega}_j' \right)}{\overline{K}_i \overline{K}_o \prod\limits_{j=1}^{N} \overline{\omega}_j'} - \frac{1}{\xi_0} = \frac{IE_r}{T_d} \tag{27}$$

The normalized integral of the error $\overline{IE}_d$ for the disturbance step is as follows:

$$\overline{IE}_d = \frac{\overline{\omega}_b^{\lambda-1}}{\overline{K}_p \overline{K}_i} = \frac{IE_d}{K_s T_d^2} \tag{28}$$

2.3. Tuning the FOPI Controller Parameters

The methods for calculating the parameters of a fractional order controller can be divided into two groups: analytical and optimization methods. In some works, the analytical calculation of the parameters of the fractional-order PI/PID controller is performed by first calculating the gains and time constants of the integer order controller. These values are then used in the fractional-order controller, whereby the orders of the integrator or the derivative terms are additionally tuned [12,18]. Such a tuning approach can provide some improvement in control performance compared to an integer order PI/PID controller, but cannot be considered optimal.

For the analytical calculation of the parameters of fractional-order PI/PID controller, methods based on the frequency responses of the open or closed loop are used instead. With such methods, the gains and time constants of a fractional-order PI/PID controller can be calculated for specific orders of the integral and derivative components [7,19]. In most cases, however, the orders of the integral and derivative components are calculated analytically [6,11,13,20–22].

Optimization methods tune the controller parameters, including the order of the integral and derivative components, to minimize the value of the optimization function. The optimization function is based on integral criteria to evaluate the quality of control [23], or on phase and gain margin requirements [24]. The approaches use different types of optimization algorithms to tune the parameters of a fractional-order PI/PID controller, including the gray wolf algorithm [25], particle swarm optimization [26], or Ant-Lion optimization [23].

The above-mentioned works, in which the FOPID controller is tuned by analytical or optimization methods, are based on control loop transfer functions that include a fractional-order integrodifferentiation operator s^{μ}. In a real control loop, this operator is replaced by a component whose properties are more or less similar to an ideal fractional order integrodifferentiator.

When an integrodifferentiator is approximated using Oustaloup's method, the lower and upper bounds of the frequency band are specified, which define the region of close similarity between the frequency characteristics of the approximated and ideal integrodifferentiators. As shown in [27], the lower limit of the frequency band of the Oustaloup integrator (ω_b) influences the disturbance rejection of the system. Increasing ω_b increases the disturbance rejection. In other words, when the parameters of the controller are calculated analytically for an ideal integrodifferentiator, its characteristics are changed by changing ω_b. Such a change causes the change in the control loop response compared to the original specifications.

Therefore, this paper presents a method to tune the FOPI controller for a normalized control loop in Figure 2, where the fractional order integrator is approximated by a rational transfer function (14). The method for calculating the controller parameters is a combination of analytical and optimization methods. The normalized controller gains $\overline{K}_p$ and $\overline{K}_i$ are calculated analytically for the given values of $\overline{\omega}_b$, $\overline{\omega}_h$ and N. The last four parameters are calculated by the optimization algorithm in such a way that the value of the integral of the absolute error is minimized at the disturbance step, whereby the control signal must not exhibit excessive deviations.

2.3.1. Calculation of the Normalized Gains of the FOPI Controller

The multiple dominant pole method (MDPM) presented in [28], which is based on the closed loop transfer function, was used to derive the formulas to calculate the normalized gains of the FOPI controller. The characteristic polynomial of the transfer function can also contain exponential terms, so that the method can also be applied to systems with transport delays. With this method, the controller parameters $\overline{K}_p$ and $\overline{K}_i$ are calculated in such a way that the transfer function of the closed control loop has a double dominant real pole $-\xi_0$. The normalized controller parameters can be calculated using the following equations:

$$\begin{aligned} 0 &= N_O(\xi)\big|_{\xi=-\xi_0} \\ 0 &= \left.\frac{dN_O(\xi)}{d\xi}\right|_{\xi=-\xi_0} \end{aligned} \tag{29}$$

After substituting for $N_O(\xi)$ in (25), the system of Equation (29) has the form

$$\begin{aligned} 0 &= -\xi_0 e^{-\xi_0} N_I(\xi_0) + \overline{K}_p N_I(\xi_0) + \overline{K}_p \overline{K}_i M_I(\xi_0) \\ 0 &= A(\xi_0) + \overline{K}_p B(\xi_0) + \overline{K}_p \overline{K}_i C(\xi_0) \end{aligned} \tag{30}$$

where the following values apply for $A(\xi_0)$, $B(\xi_0)$, $C(\xi_0)$, $M_I(\xi_0)$ and $N_I(\xi_0)$:

$$\begin{aligned} A(\xi_0) &= \left.\frac{d\left(\xi e^{\xi} N_I(\xi)\right)}{d\xi}\right|_{\xi=-\xi_0} \\ B(\xi_0) &= \left.\frac{dN_I(\xi)}{d\xi}\right|_{\xi=-\xi_0} \\ C(\xi_0) &= \left.\frac{dM_I(\xi)}{d\xi}\right|_{\xi=-\xi_0} \\ M_I(\xi_0) &= M_I(\xi)\big|_{\xi=-\xi_0} \\ N_I(\xi_0) &= N_I(\xi)\big|_{\xi=-\xi_0} \end{aligned} \tag{31}$$

The solution of (30) is as follows:

$$\begin{aligned} \overline{K}_p &= \frac{M_I(\xi_0)A(\xi_0) + \xi_0 e^{-\xi_0} N_I(\xi_0)C(\xi_0)}{N_I(\xi_0)C(\xi_0) - M_I(\xi_0)B(\xi_0)} \\ \overline{K}_i &= -\frac{N_I(\xi_0)\left(A(\xi_0) + \xi_0 e^{-\xi_0} B(\xi_0)\right)}{M_I(\xi_0)A(\xi_0) + \xi_0 e^{-\xi_0} N_I(\xi_0)C(\xi_0)} \end{aligned} \tag{32}$$

Expression (32) is used to calculate the values of $\overline{K}_p$ and $\overline{K}_i$ for the specified dominant pole $-\xi_0$. The values of ξ_0 are determined together with the order of the fractional order integrator λ and the lower limit of the frequency band of the approximated integrator $\overline{\omega}_b$ using an algorithm described in Section 2.3.2.

2.3.2. The Optimum Values for ξ_0, λ, and $\overline{\omega}_b$

The transfer function of the closed control loop $G_r(\xi)$ is of order $N+2$, where $N \geq 1$. Therefore, in addition to the $-\xi_0$ pole, it has other poles, the number and position of which depend on the order and other parameters of the approximated integrator. The additional poles can have a considerable influence on the closed control loop. An unsuitable setting of the parameters of the approximated integrator can lead to oscillations or even instability of the control loop. Therefore, the algorithm proposed here searches for the values of ξ_0, λ, and $\overline{\omega}_b$ for the specified N and $\overline{\omega}_h$ in order to minimize the value of the integral of absolute error (IAE) at the disturbance step:

$$IAE = \int_0^\infty |\omega^*(t) - \omega(t)| dt \tag{33}$$

In addition, the deviations of the control signal from the one-pulse function (1P) are evaluated at the setpoint and disturbance steps, where the 1P function is defined in [29]. The deviations can be calculated using the following expression [30]:

$$TV_1(\overline{U}^*) = \sum_c \left|\overline{U}^*_{c+1} - \overline{U}^*_c\right| - \left|2\overline{U}^*_{max} - \overline{U}^*_\infty - \overline{U}^*_0\right| \tag{34}$$

where c is the number of samples. $\overline{U}^*_0$, $\overline{U}^*_\infty$, $\overline{U}^*_{\max}$ are the initial, the final steady state, and the maximum value of the control signal for the setpoint ω^* and the disturbance $\overline{M}_L$ steps, respectively.

The optimization objective is expressed by the function Γ_d in (35), where IAE_d is the integral of the absolute error at the disturbance step, ε_r is the maximum allowable shape deviation at the setpoint or disturbance step, $TV_{1r}(\overline{U}^*)$ and $TV_{1d}(\overline{U}^*)$ are the shape deviations of the control signal from the 1P function at the setpoint and disturbance step, respectively, and $k \in \mathbb{N}$ is the cycle number of the optimization algorithm given below. The values of IAE_d, $TV_{1r}(\overline{U}^*)$, and $TV_{1d}(\overline{U}^*)$ are determined by a simulation in which the setpoint step is executed first and then the disturbance step.

$$\begin{gathered}\Gamma_d = \min\left(IAE_d(\overline{\omega}_b, \xi_0, \lambda)\right) \cap \left(TV_{1r}(\overline{U}^*) \leq \varepsilon_r\right) \cap \left(TV_{1d}(\overline{U}^*) \leq \varepsilon_r\right) \\ \overline{\omega}_b \in \langle \overline{\omega}^k_{b,\min}, \overline{\omega}^k_{b,\max} \rangle = \overline{\Omega}^k_b, \quad \xi_0 \in \langle \xi^k_{0,\min}, \xi^k_{0,\max} \rangle = \Xi^k_0 \\ \lambda \in \langle \lambda^k_{\min}, \lambda^k_{\max} \rangle = \Lambda^k \end{gathered} \tag{35}$$

The authors of the article have proposed an algorithm that searches the three-dimensional space $\mathbf{H}^k = \overline{\Omega}^k_b \times \Xi^k_0 \times \Lambda^k$ in several cycles. When searching the space $\mathbf{H}^k$, the values of the variables $\overline{\omega}_b$, ξ_0 and λ are changed with the step ΔP^k_n according to (36), where the index n is one of the variables in Table 1. N_{oP} is the number of all tested values of a particular variable (i.e., the values of ω_b, ξ_0, or λ) applied in one cycle of the algorithm.

$$\begin{gathered}\Delta P^k_n = \begin{cases} \dfrac{P^1_{n,\max} - P^1_{n,\min}}{N_{oP} - 1} & k = 1 \\[2ex] \dfrac{P^{k-1}_{n,\max} - P^{k-1}_{n,\min}}{2^{1/3} \times (N_{oP} - 1)} & k > 1 \end{cases} \\ n \in \{1,2,3\}, N_{oP} \in \mathbb{N}, N_{oP} \geq 5 \end{gathered} \tag{36}$$

Table 1. Relations between the variables of the optimization function Γ_d and the variables of (36).

Variable in (36)	**Corresponding Variable**		
	$\boldsymbol{n=1}$	$\boldsymbol{n=2}$	$\boldsymbol{n=3}$
P_n	$\overline{\omega}_b$	ξ_0	λ
ΔP_n^k	$\Delta\overline{\omega}_b^k$	$\Delta\xi_0^k$	$\Delta\lambda^k$
$P_{n,\min}^k$	$\overline{\omega}_{b,\min}^k$	$\xi_{0,\min}^k$	$\lambda_{\min}^k$
$P_{n,\max}^k$	$\overline{\omega}_{b,\max}^k$	$\xi_{0,\max}^k$	$\lambda_{\max}^k$

The size ("volume") of the searched space is halved with each cycle k, as shown in Figure 3, but the number of tested values of the variables (for which IAE_d, TV_{1r}, and TV_{1d} are calculated) remains unchanged. The limits of the searched interval in the k-th cycle are

$$\begin{array}{l} P_{n,\min}^k = P_{n,opt}^{k-1} - \frac{N_{oP}}{2}\Delta P_n^k \ \cap \ P_{n,\min}^k \geq P_{n,\min}^1 \\ P_{n,\max}^k = P_{n,opt}^{k-1} + \frac{N_{oP}}{2}\Delta P_n^k \ \cap \ P_{n,\max}^k \leq P_{n,\max}^1 \end{array} \tag{37}$$

where $P_{n,opt}^{k-1}$ is the value of the parameter P_n that satisfies (35) in the $k-1$ cycle. The Formula (37) is valid for $k > 1$. The values of the parameter P_n applied in cycle k are symmetrically distributed around a central value $P_{n,opt}^{k-1}$. The user specifies the range of parameter changes P_n in the first cycle of the algorithm ($k = 1$).

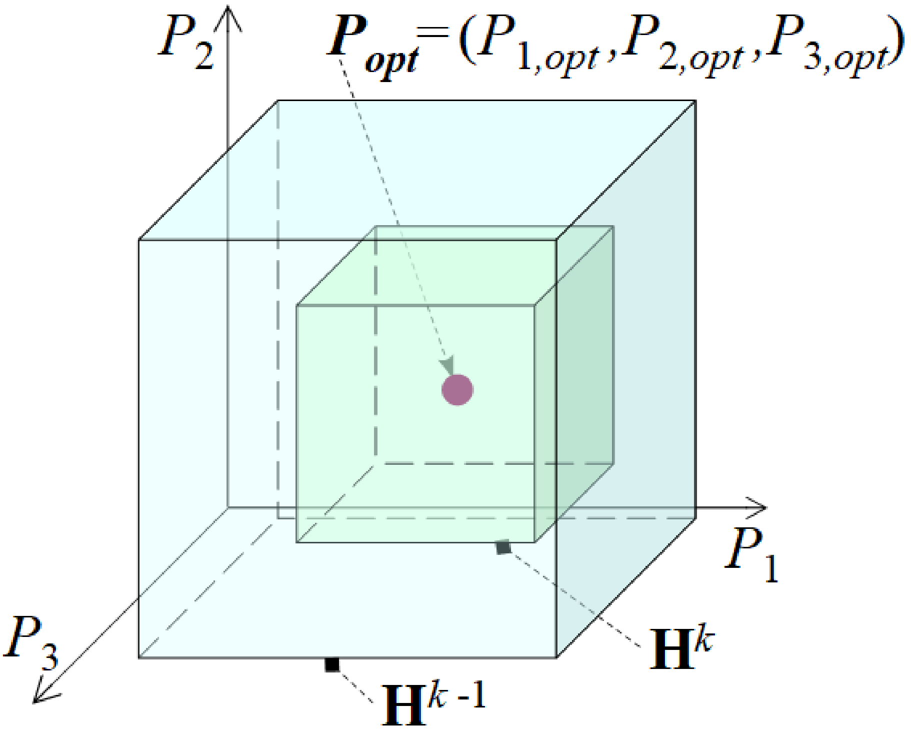

Figure 3. The space **H** in the k-th cycle of the algorithm.

Figure 4 shows the flowchart of the algorithm for selecting the values of $\overline{\omega}_b$, ξ_0, and λ so that expression (35) applies. The algorithm is executed in k_{max} cycles. The innermost loop of the algorithm is the loop in which the value of the parameter $P_3 = \lambda$ is calculated. This loop is run through N_{oP}^3 times in one cycle. For $N_{oP} = 19$ and $k_{max} = 20$, the innermost loop of the algorithm, in which the parameters of the controller and the fractional order integrator are calculated, is run through 137,180 times. The output variables of the algorithm are: $\overline{\omega}_{b,opt}$, $\xi_{0,opt}$, λ_{opt}, $\overline{K}_{p,opt}$, $\overline{K}_{i,opt}$, $\overline{IAE}_{r,opt}$, $\overline{IAE}_{d,opt}$.

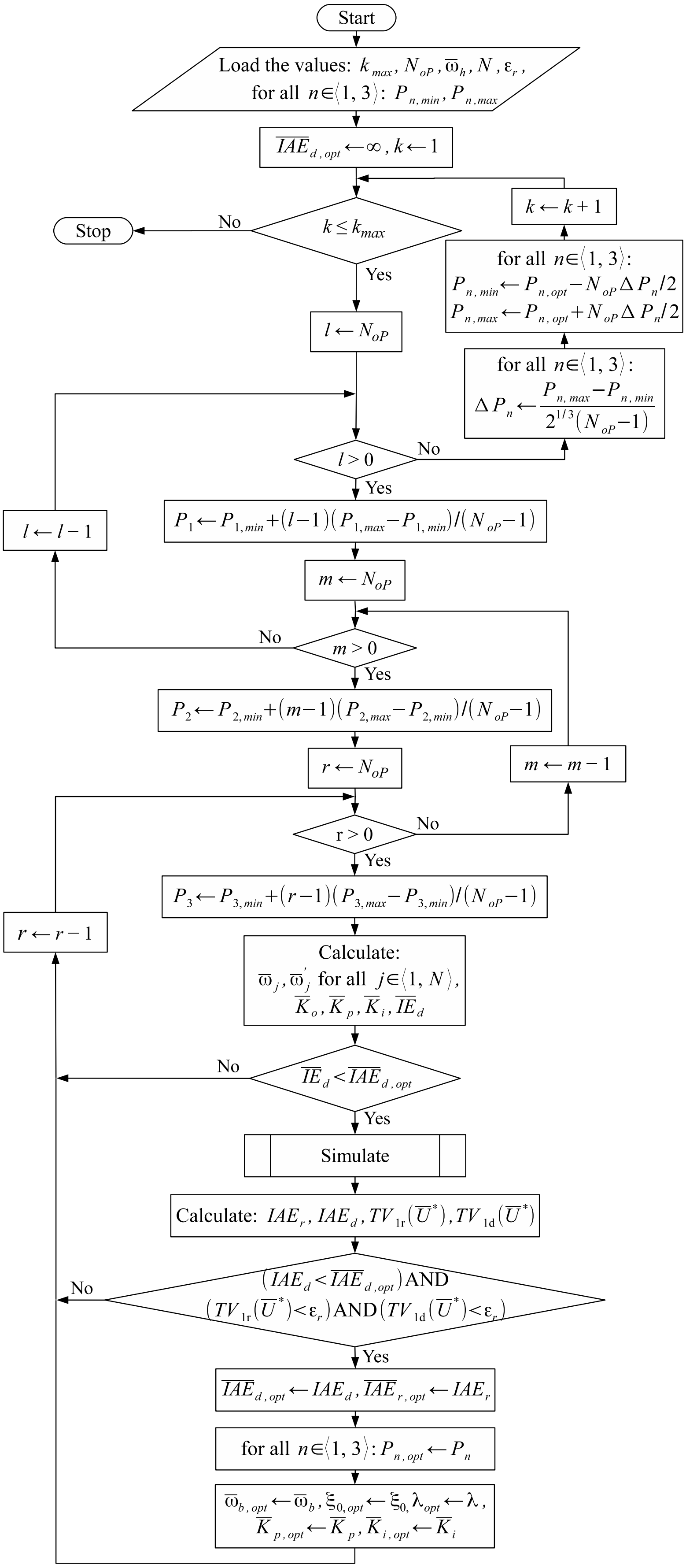
Start
Load the values: k_{max}, N_{oP}, $\overline{\varpi}_h$, N, ε_r,
for all $n\in\langle 1, 3\rangle$: $P_{n,min}$, $P_{n,max}$
$\overline{IAE}_{d,opt}\leftarrow\infty$, $k\leftarrow 1$
$k\leftarrow k+1$
No
Stop
$k\le k_{max}$
Yes
for all $n\in\langle 1, 3\rangle$:
$P_{n,min}\leftarrow P_{n,opt}-N_{oP}\Delta P_n/2$
$P_{n,max}\leftarrow P_{n,opt}+N_{oP}\Delta P_n/2$
$l\leftarrow N_{oP}$
for all $n\in\langle 1, 3\rangle$:
$\Delta P_n\leftarrow\frac{P_{n,max}-P_{n,min}}{2^{1/3}(N_{oP}-1)}$
$l>0$
No
Yes
$l\leftarrow l-1$
$P_1\leftarrow P_{1,min}+(l-1)(P_{1,max}-P_{1,min})/(N_{oP}-1)$
$m\leftarrow N_{oP}$
No
$m>0$
Yes
$P_2\leftarrow P_{2,min}+(m-1)(P_{2,max}-P_{2,min})/(N_{oP}-1)$
$r\leftarrow N_{oP}$
$m\leftarrow m-1$
r > 0
No
Yes
$P_3\leftarrow P_{3,min}+(r-1)(P_{3,max}-P_{3,min})/(N_{oP}-1)$
$r\leftarrow r-1$
Calculate:
$\overline{\varpi}_j$, $\overline{\varpi}_j'$ for all $j\in\langle 1, N\rangle$,
$\overline{K}_o$, $\overline{K}_p$, $\overline{K}_i$, $\overline{IE}_d$
No
$\overline{IE}_d<\overline{IAE}_{d,opt}$
Yes
Simulate
Calculate: IAE_r, IAE_d, $TV_{1r}(\overline{U}^*)$, $TV_{1d}(\overline{U}^*)$
No
$(IAE_d<\overline{IAE}_{d,opt})$ AND
$(TV_{1r}(\overline{U}^*)<\varepsilon_r)$ AND $(TV_{1d}(\overline{U}^*)<\varepsilon_r)$
Yes
$\overline{IAE}_{d,opt}\leftarrow IAE_d$, $\overline{IAE}_{r,opt}\leftarrow IAE_r$
for all $n\in\langle 1, 3\rangle$: $P_{n,opt}\leftarrow P_n$
$\overline{\varpi}_{b,opt}\leftarrow\overline{\varpi}_b$, $\xi_{0,opt}\leftarrow\xi_0$, $\lambda_{opt}\leftarrow\lambda$,
$\overline{K}_{p,opt}\leftarrow\overline{K}_p$, $\overline{K}_{i,opt}\leftarrow\overline{K}_i$

Figure 4. Flowchart of algorithm using symmetrically distributed points in 3D space to select $\overline{\omega}_b$, ξ_0, and λ values.

Matlab R 2018b with Symbolic Math Toolbox was used to compute the parameters of the normalized FOPI controller using the proposed optimization algorithm. Simulink V 9.2 and FOMCON Toolbox V 1.50.3 were used to simulate the FOPI controller.

2.4. PI Controller Tuning

To compare the performance of the FOPI controller with that of the PI controller, the tuning of the integer order PI controller that minimizes the IAE at the disturbance step is presented below.

The structure of the control loop is the same as in Figure 2. The polynomials in the numerator and denominator of the integrator are as follows

$$M_I(\xi)|_{\lambda=1} = 1, \quad N_I(\xi)|_{\lambda=1} = \xi \tag{38}$$

and the transfer functions $G_r(\xi)$ and $G_d(\xi)$ (23) become:

$$G_{1r}(\xi) = \frac{\overline{K}_p\xi + \overline{K}_p\overline{K}_i}{e^{\xi} + \overline{K}_p\xi + \overline{K}_p\overline{K}_i} \tag{39}$$

$$G_{1d}(\xi) = \frac{-e^{\xi}\xi}{e^{\xi} + \overline{K}_p\xi + \overline{K}_p\overline{K}_i} \tag{40}$$

The expression for calculating the gains of a PI controller ($\lambda = 1$) is now as follows:

$$\overline{K}_p|_{\lambda=1} = \xi_0 e^{-\xi_0}(2-\xi_0), \quad \overline{K}_i|_{\lambda=1} = \xi_0\frac{1-\xi_0}{2-\xi_0} \tag{41}$$

where $-\xi_0$ is the double dominant real pole of the transfer functions $G_{1r}(\xi)$ and $G_{1d}(\xi)$.

The reference speed filter $F_I(\xi)$

$$F_I(\xi)|_{\lambda=1} = \frac{\xi_0^{-1}\xi + 1}{\overline{K}_i^{-1}\xi + 1} \tag{42}$$

compensates for the zero in the numerator of the transfer function $G_{1r}(\xi)$ and a real pole $-\xi_0$.

The $\overline{IE}$ values at the setpoint and disturbance steps when applying the $F_I(\xi)$ filter are (42):

$$\overline{IE}_r|_{\lambda=1} = \frac{1}{\overline{K}_i} - \frac{1}{\xi_0} = \frac{1}{\xi_0(1-\xi_0)} \tag{43}$$

$$\overline{IE}_d|_{\lambda=1} = \frac{1}{\overline{K}_p\overline{K}_i} = \frac{e^{\xi_0}}{\xi_0^2(1-\xi_0)} \tag{44}$$

The minimum values of the $\overline{IE}$ and the corresponding values of ξ_0, $\overline{K}_p$, $\overline{K}_i$ are

$$\begin{array}{llll} \overline{IE}_{r,min} = 4, & \xi_0 = 0.5, & \overline{K}_p = 0.4549, \overline{K}_i = 0.1667, & \lambda = 1 \\ \overline{IE}_{d,min} = 12.6387, & \xi_0 = 0.5858, & \overline{K}_p = 0.4612, \overline{K}_i = 0.1716, & \lambda = 1 \end{array} \tag{45}$$

and since there are no overshoots in the setpoint and disturbance steps, the $\overline{IAE}$ values are identical to the $\overline{IE}$ values:

$$\overline{IAE}_{r,min}|_{\lambda=1} = \overline{IE}_{r,min}|_{\lambda=1}, \quad \overline{IAE}_{d,min}|_{\lambda=1} = \overline{IE}_{d,min}|_{\lambda=1} \tag{46}$$

A comparison of the achieved results with the fully analytical PI controller design using the triple real dominant pole method in [31] shows that the new methodology gives two optimal controller parameter sets instead of a single set of parameters—one optimal for setpoint steps, the other for disturbance steps. However, this only confirms other results of [31] obtained by the experimental performance portrait method (PPM). PPM showed that for different weighting of setpoint and disturbance steps, there will also be different optimal settings of the controller (see Table 1 in [31]).

2.5. Experimental Workstation

The performance of the speed control loop with the FOPI controller tuned by the proposed method described in Section 2.3 was verified on a workstation whose block diagram is shown in Figure 5. The workstation consists of two servo drives, a motion controller, and a personal computer running Matlab R 2023a software (Mathworks, Natick, MA, USA) and Automation Studio V 4.6 (Bernecker & Reiner, Eggelsberg, Austria).

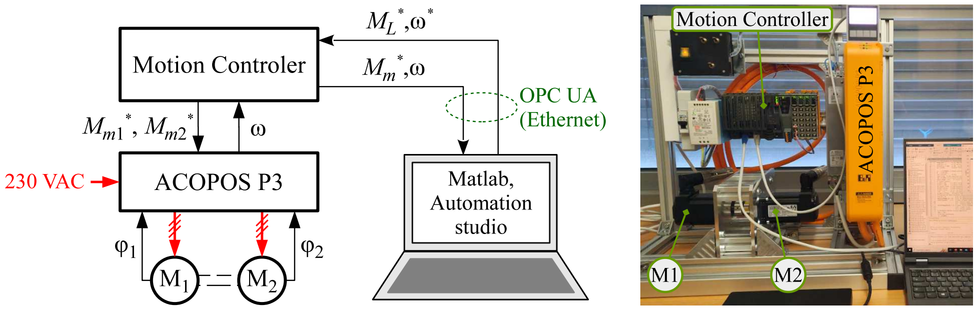

Figure 5. Block diagram and a photography of the workstation.

The servo drives contain two permanent magnet synchronous motors (M_1, M_2) of type 8LVA23 with a rated power $P_N = 400$ W and a rated speed $n_N = 3000$ rpm. The shafts of the motors are connected via a rigid coupling. One of the motors is speed-controlled, while the second motor generates the load torque M_L. Absolute position sensors with the current rotor positions φ_1 and φ_2 are integrated in the motors. The information about the current rotor position is processed by an ACOPOS P3 frequency inverter (Bernecker & Reiner) to control the torque of the motors and calculate the actual speed ω. The speed controller is implemented in a motion control unit of type X20CP1586 (Bernecker & Reiner), where the control algorithm is executed with a sampling period T_s.

The reference speed ω^* and the load torque M_L^* are sent from Matlab to the motion controller via OPC UA. Matlab receives the values for the target motor torque M_m^* and the actual speed ω from the motion controller. The motion controller sends the target motor torque values M_{m1}^* and M_{m2}^* to the frequency inverter and receives the actual speed ω from it.

The speed controller was implemented in the motion controller using Matlab and the Automation Studio Target for Simulink Toolbox. Using the toolbox, Matlab compiles the program for the control algorithm in C language and saves it in the Automation Studio project. This loads the program into the memory of the motion controller and executes it.

The basic form of the fractional order PI controller was created in Simulink. The integrator of the controller had the structure of a sequentially connected first-order system with the first-order integrator according to (14), which was then transformed into a discrete form using Tustin's method [32].

The controlled system parameters used to calculate the process and controller parameters are listed in Table 2.

Table 2. Parameters of the controlled system.

Symbol	Value	Unit	Description
J	6.5×10^{-6}	kgm^2	Moment of inertia
K_s	15,385	kg^{-1}m^{-2}	System gain
T_{GM}	5	ms	Torque generator transport delay
T_s	0.4	ms	Sampling period

The value of the moment of inertia J was obtained during the autotuning of the torque generators in Automation Studio software. The system gain K_s was calculated from the moment of inertia according to (8). The torque generator transport delay T_{GM} has been identified from the speed responses to step changes in the reference torque M_m^*. The sampling period $T_s = 0.4$ ms corresponds to the minimum possible program cycle length for the X20CP1586 motion controller.

3. Results

The results fall into two groups.

The first group is represented by tables with calculated optimal values of the normalized FOPI controller parameters. The calculations of the parameter values in these tables were performed using the method presented in Section 2.3. The controller parameters have been calculated for the specified values of the upper limit of the frequency band of the approximated integrator $\overline{\omega}_h$ and for the selected orders of the integrator transfer function N. These results are presented below in Section 3.1.

Experimental results on a speed actuator with FOPI and PI controller represent the second group of achieved results. The parameters of the FOPI controller were calculated using the results in Section 3.1. The description of the experiments and their results are in Section 3.2.

3.1. Optimization Results

A simulation model was created in Simulink based on Figure 2 and the methods presented in Sections 2.3.1 and 2.3.2. The parameters of the normalized FOPI controller and the normalized fractional-order integrator, approximated by Oustaloup's method, were calculated for the chosen values of $\overline{\omega}_h$ and N. During the simulations, unit steps of the reference speed ω^* and the normalized disturbance $\overline{M}_L$ were applied. The allowable shape deviation of the control signal from the 1P signal for the reference and disturbance steps, the number of algorithm cycles, and the number of different values of the variables in an algorithm cycle were as follows: $\varepsilon_r = 1 \times 10^{-6}$, $k_{max} = 20$, $N_{oP} = 19$, respectively. The normalized parameters, including the normalized $\overline{IAE}$ values, are shown in Tables 3–13. The calculation time for the values of one row of the table took between two and eight hours, depending on the values of $\overline{\omega}_h$ and N.

Table 3. Optimized controller parameters and achieved $\overline{IAE}$ values, $\overline{\omega}_h = 0.2$.

N	$\overline{\omega}_b$	ξ_0	λ	$\overline{K}_p$	$\overline{K}_i$	$\overline{IAE}_r$	$\overline{IAE}_d$
1	0.19904	0.58542	1.0430	0.46118	0.16015	9.1293	12.6327
2	0.19954	0.58517	1.0903	0.46118	0.14840	14.1279	12.6328
3	0.19931	0.58510	1.0619	0.46118	0.15533	19.1416	12.6331
5	0.19935	0.58496	1.1298	0.46120	0.13930	29.1419	12.6382

$\overline{\omega}^1_{b,min} = 0.1$, $\overline{\omega}^1_{b,max} = 0.2$, $\xi^1_{0,min} = 0.2$, $\xi^1_{0,max} = 0.8$, $\lambda^1_{min} = 0.3$, $\lambda^1_{max} = 2$.

Table 4. Optimized controller parameters and achieved $\overline{IAE}$ values, $\overline{\omega}_h = 0.25$.

N	$\overline{\omega}_b$	ξ_0	λ	$\overline{K}_p$	$\overline{K}_i$	$\overline{IAE}_r$	$\overline{IAE}_d$
1	0.24075	0.27282	1.1908	0.62507	0.14912	5.5068	8.1764
2	0.23876	0.27227	1.1578	0.62593	0.15600	9.6124	8.1700
3	0.24390	0.27314	1.2597	0.62616	0.13527	13.5996	8.1836
5	0.23122	0.26760	1.0828	0.62736	0.17065	22.2419	8.2751

$\overline{\omega}^1_{b,min} = 0.15$, $\overline{\omega}^1_{b,max} = 0.25$, $\xi^1_{0,min} = 0.1$, $\xi^1_{0,max} = 0.9$, $\lambda^1_{min} = 0.3$, $\lambda^1_{max} = 2$.

Table 5. Optimized controller parameters and achieved $\overline{IAE}$ values, $\overline{\omega}_h = 0.3$.

N	$\overline{\omega}_b$	ξ_0	λ	$\overline{\mathbf{K}}_\mathbf{p}$	$\overline{\mathbf{K}}_\mathbf{i}$	$\overline{\mathbf{IAE}}_\mathbf{r}$	$\overline{\mathbf{IAE}}_\mathbf{d}$
1	0.27968	0.32034	1.0834	0.60629	0.18876	5.0844	7.8575
2	0.28473	0.32234	1.1002	0.60617	0.18499	8.4988	7.8632
3	0.27806	0.31896	1.0658	0.60819	0.19173	12.0398	7.8838
5	0.28868	0.32531	1.1438	0.60412	0.17630	18.6460	7.8586

$\overline{\omega}^1_{b,min} = 1 \times 10^{-4}$, $\overline{\omega}^1_{b,max} = 0.3$, $\xi^1_{0,min} = 0.1$, $\xi^1_{0,max} = 0.9$, $\lambda^1_{min} = 0.1$, $\lambda^1_{max} = 2$.

Table 6. Optimized controller parameters and achieved $\overline{IAE}$ values, $\overline{\omega}_h = 0.5$.

N	$\overline{\omega}_b$	ξ_0	λ	$\overline{\mathbf{K}}_\mathbf{p}$	$\overline{\mathbf{K}}_\mathbf{i}$	$\overline{\mathbf{IAE}}_\mathbf{r}$	$\overline{\mathbf{IAE}}_\mathbf{d}$
1	0.49136	0.42308	1.7353	0.60432	0.12606	4.3456	7.7850
2	0.48887	0.42156	1.5779	0.60506	0.14037	6.3713	7.7865
3	0.48093	0.42119	1.3	0.60365	0.17067	8.4360	7.7925
5	0.48363	0.41924	1.3892	0.60469	0.15948	12.4959	7.8157

$\overline{\omega}^1_{b,min} = 1 \times 10^{-4}$, $\overline{\omega}^1_{b,max} = 0.5$, $\xi^1_{0,min} = 0.1$, $\xi^1_{0,max} = 0.9$, $\lambda^1_{min} = 0.1$, $\lambda^1_{max} = 2$.

Table 7. Optimized controller parameters and achieved $\overline{IAE}$ values, $\overline{\omega}_h = 1$.

N	$\overline{\omega}_b$	ξ_0	λ	$\overline{\mathbf{K}}_\mathbf{p}$	$\overline{\mathbf{K}}_\mathbf{i}$	$\overline{\mathbf{IAE}}_\mathbf{r}$	$\overline{\mathbf{IAE}}_\mathbf{d}$
1	0.40311	0.44050	1.0811	0.63654	0.19193	4.0884	7.6043
2	0.83559	0.52196	1.9890	0.65084	0.18033	4.8221	7.1337
3	0.83715	0.52363	1.9896	0.64986	0.18092	5.9138	7.1334
5	0.83348	0.51830	2.0	0.65323	0.17888	8.1150	7.1329

$\overline{\omega}^1_{b,min} = 1 \times 10^{-4}$, $\overline{\omega}^1_{b,max} = 1$, $\xi^1_{0,min} = 0.1$, $\xi^1_{0,max} = 0.9$, $\lambda^1_{min} = 0.1$, $\lambda^1_{max} = 2$.

Table 8. Optimized controller parameters and achieved $\overline{IAE}$ values, $\overline{\omega}_h = 2$.

N	$\overline{\omega}_b$	ξ_0	λ	$\overline{\mathbf{K}}_\mathbf{p}$	$\overline{\mathbf{K}}_\mathbf{i}$	$\overline{\mathbf{IAE}}_\mathbf{r}$	$\overline{\mathbf{IAE}}_\mathbf{d}$
1	1.0712	0.52406	1.9963	0.67959	0.21140	3.6584	7.4545
2	0.96845	0.49373	1.9124	0.73147	0.19081	4.3024	6.9584
3	1.1133	0.56681	2.0	0.71028	0.23658	4.7885	6.6255
5	1.1377	0.58334	2.0	0.70271	0.24507	6.1010	6.6065

$\overline{\omega}^1_{b,min} = 1 \times 10^{-4}$, $\overline{\omega}^1_{b,max} = 2$, $\xi^1_{0,min} = 0.1$, $\xi^1_{0,max} = 0.9$, $\lambda^1_{min} = 0.1$, $\lambda^1_{max} = 2$.

Table 9. Optimized controller parameters and achieved $\overline{IAE}$ values, $\overline{\omega}_h = 3$.

N	$\overline{\omega}_b$	ξ_0	λ	$\overline{\mathbf{K}}_\mathbf{p}$	$\overline{\mathbf{K}}_\mathbf{i}$	$\overline{\mathbf{IAE}}_\mathbf{r}$	$\overline{\mathbf{IAE}}_\mathbf{d}$
1	1.2077	0.55501	1.9913	0.68786	0.23922	3.5730	7.3271
2	1.1599	0.53677	1.9913	0.71464	0.22907	4.0651	7.0761
3	1.0413	0.52033	1.8448	0.74531	0.20657	4.6112	6.7212
5	1.2261	0.58440	2.0	0.73461	0.25918	5.4803	6.4695

$\overline{\omega}^1_{b,min} = 1 \times 10^{-4}$, $\overline{\omega}^1_{b,max} = 2$, $\xi^1_{0,min} = 0.1$, $\xi^1_{0,max} = 0.9$, $\lambda^1_{min} = 0.1$, $\lambda^1_{max} = 2$.

Table 10. Optimized controller parameters and achieved $\overline{IAE}$ values, $\overline{\omega}_h = 5$.

N	$\overline{\omega}_b$	ξ_0	λ	$\overline{\mathbf{K}}_\mathbf{p}$	$\overline{\mathbf{K}}_\mathbf{i}$	$\overline{\mathbf{IAE}}_\mathbf{r}$	$\overline{\mathbf{IAE}}_\mathbf{d}$
1	1.3231	0.57339	2.0	0.70114	0.26177	3.5106	7.2091
2	1.3004	0.56600	2.0	0.71213	0.25719	3.8814	7.0998
3	1.2405	0.54600	1.9913	0.73529	0.24315	4.2876	6.9254
5	1.1330	0.55400	1.8168	0.75484	0.22603	5.1232	6.4903

$\overline{\omega}^1_{b,min} = 1 \times 10^{-4}$, $\overline{\omega}^1_{b,max} = 2$, $\xi^1_{0,min} = 0.1$, $\xi^1_{0,max} = 0.9$, $\lambda^1_{min} = 0.1$, $\lambda^1_{max} = 2$.

Table 11. Optimized controller parameters and achieved $\overline{IAE}$ values, $\overline{\omega}_h = 10$.

N	$\overline{\omega}_b$	ξ_0	λ	$\overline{\mathbf{K}}_\mathbf{p}$	$\overline{\mathbf{K}}_\mathbf{i}$	$\overline{\mathbf{IAE}}_\mathbf{r}$	$\overline{\mathbf{IAE}}_\mathbf{d}$
1	1.4152	0.58932	2.0	0.71148	0.27957	3.4653	7.1150
2	1.4104	0.58849	2.0	0.71473	0.27899	3.7222	7.0728
3	1.3693	0.57426	1.9963	0.72664	0.26902	4.0140	6.9968
5	0.9369	0.51061	1.6065	0.75571	0.18844	4.8956	6.7499

$\overline{\omega}^1_{b,min} = 1 \times 10^{-4}$, $\overline{\omega}^1_{b,max} = 2$, $\xi^1_{0,min} = 0.1$, $\xi^1_{0,max} = 0.9$, $\lambda^1_{min} = 0.1$, $\lambda^1_{max} = 2$.

Table 12. Optimized controller parameters and achieved $\overline{IAE}$ values, $\overline{\omega}_h = 20$.

N	$\overline{\omega}_b$	ξ_0	λ	$\overline{\mathbf{K}}_\mathbf{p}$	$\overline{\mathbf{K}}_\mathbf{i}$	$\overline{\mathbf{IAE}}_\mathbf{r}$	$\overline{\mathbf{IAE}}_\mathbf{d}$
1	1.4573	0.59854	1.9956	0.71654	0.28731	3.4434	7.0670
2	1.4381	0.58726	2.0	0.72267	0.28225	3.6287	7.0504
3	1.4666	0.60071	2.0	0.71949	0.29073	3.8349	7.0115
5	1.4094	0.58031	2.0	0.73801	0.27741	4.2996	6.8839

$\overline{\omega}^1_{b,min} = 1 \times 10^{-4}$, $\overline{\omega}^1_{b,max} = 2$, $\xi^1_{0,min} = 0.1$, $\xi^1_{0,max} = 0.9$, $\lambda^1_{min} = 0.1$, $\lambda^1_{max} = 2$.

Table 13. Optimized controller parameters and achieved $\overline{IAE}$ values, $\overline{\omega}_h = 50$.

N	$\overline{\omega}_b$	ξ_0	λ	$\overline{\mathbf{K}}_\mathbf{p}$	$\overline{\mathbf{K}}_\mathbf{i}$	$\overline{\mathbf{IAE}}_\mathbf{r}$	$\overline{\mathbf{IAE}}_\mathbf{d}$
1	1.2936	0.59928	1.9250	0.72134	0.24988	3.4322	7.0396
2	1.4773	0.60060	1.9927	0.72146	0.29036	3.5456	7.0321
3	1.4601	0.58954	2.0	0.72784	0.28607	3.7038	7.0127
5	1.4399	0.58346	2.0	0.73578	0.28190	4.0470	6.9422

$\overline{\omega}^1_{b,min} = 1 \times 10^{-4}$, $\overline{\omega}^1_{b,max} = 2$, $\xi^1_{0,min} = 0.1$, $\xi^1_{0,max} = 0.9$, $\lambda^1_{min} = 0.1$, $\lambda^1_{max} = 2$.

3.2. Laboratory Experiments

Experiments were performed with the servo system described in Section 2.5. The servo system consists of a controlled drive with a torque generator and a load drive that generates load torque steps.

3.2.1. Description of the Experiments

One experiment with an integer order PI controller and a series of experiments with a FOPI speed controller were performed on the workstation. Each experiment evaluated the response of the actual speed ω to a step change in:

- the reference speed from the value ω_1^* to the value ω_2^* at a constant load torque $M_L = M_{L,1}$ at time $t = t_1$,
- the load torque from the value $M_{L,1}$ to the value $M_{L,2}$ at a constant actual speed $\omega = \omega_2^*$ at time $t = t_2$.

The specific values for the reference speed and the load torque can be found in Table 14. The time t refers to the start of the experiment.

Table 14. Reference values and timing of the experiment.

Symbol	Value	Unit
$M_{L,1}$	0.05	Nm
$M_{L,2}$	0.2	Nm
t_1	1	s
t_2	2	s
ω_1^*	40	rad/s
ω_2^*	80	rad/s

3.2.2. Actual Speed Controller Parameters

The parameters of the speed controller and the fractional order integrator were determined from the normalized parameters given in Tables 3–13 on the basis of the parameters listed in Table 2. The formulas for converting the normalized parameters to the actual parameters are as follows

$$\begin{gathered}\omega_b = \frac{\overline{\omega}_b}{T_d}, \quad \omega_h = \frac{\overline{\omega}_h}{T_d}, \quad K_o = \left(\frac{\overline{\omega}_h}{T_d}\right)^{1-\lambda}, \quad K_p = \frac{\overline{K}_p}{K_s T_d}, \quad K_i = \frac{\overline{K}_i}{T_d^{\lambda}}, \quad s_0 = \frac{\xi_0}{T_d} \\ \omega_j = \frac{\overline{\omega}_b}{T_d}\left(\frac{\overline{\omega}_h}{\overline{\omega}_b}\right)^{\frac{2j-\lambda}{2N}}, \quad \omega_j' = \frac{\overline{\omega}_b}{T_d}\left(\frac{\overline{\omega}_h}{\overline{\omega}_b}\right)^{\frac{2j-2+\lambda}{2N}}, \quad j \in \mathbb{N},\ 1 \le j \le N\end{gathered} \tag{47}$$

Experiments with the FOPI controller were carried out with parameters calculated for:

- $N = 3$ and $\overline{\omega}_h \in \{0.3, 0.5, 1, 2, 3, 5, 10\}$.
- $\overline{\omega}_h = 5$ and $N \in \{1, 2, 3, 5\}$,

The value $N = 3$ in the first set of experiments was chosen as the average of the interval 1 to 5. The values $\overline{\omega}_h \ge 0.3$ were chosen because there is a significant decrease in the value of $\overline{IAE}_d$. For $\overline{\omega}_h = 5$ in the second set of experiments, no time constant less than twice the sampling period $T_s = 0.4$ ms appears in the transfer function of the approximated fractional-order integrator (this is not the case for $\overline{\omega}_h \ge 10$). This $\overline{\omega}_h$ setting guarantees the fastest output response for this tuning method and for this system.

The calculated controller parameters are listed in Table 15.

Table 15. PI and FOPI controller parameters used in the experiments.

$\overline{\omega}_h$	N	ω_b [rad/s]	ω_h [rad/s]	$\mathrm{K_o} \times 10^3$	λ	$\mathrm{K_p} \times 10^3$ [Nms/rad]	$\mathrm{K_i}$ [$\mathrm{s^{-1}}$]	$\mathrm{s_0}$ [rad/s]
–	–	–	–	–	1	5.7643	32.99479	112.654
0.3	3	53.473	57.692	765.81	1.0658	7.6022	52.11632	61.338
0.5	3	92.487	96.154	254.16	1.3000	7.5454	158.9838	80.998
1	3	160.990	192.308	5.4923	1.9896	8.1230	6334.704	100.698
2	3	214.096	384.615	2.6000	2.0	8.8783	8749.260	109.002
3	3	200.250	576.923	4.6495	1.8448	9.3161	3377.414	100.063
5	3	238.558	961.538	1.1040	1.9913	9.1909	8590.072	105.000
10	3	263.327	1923.08	5.3475	1.9963	9.0828	9757.242	110.435
5	1	254.442	961.538	1.0400	2.0	8.7640	9680.843	110.267
5	2	250.077	961.538	1.0400	2.0	8.9014	9511.465	108.846
5	5	217.885	961.538	3.6603	1.8168	9.4353	3189.564	106.538

In a speed control loop with an FOPI controller, the reference speed ω^* is filtered by a filter $F_I(s)$ (18), where the polynomials $M_I(s)$ and $N_I(s)$ are given in (14). For an integer order PI controller, the transfer function of the speed setpoint filter is given by (48).

$$F_I(s)|_{\lambda=1} = \frac{s_0^{-1}s + 1}{K_i^{-1}s + 1} \tag{48}$$

3.2.3. Experimental Results

Figures 6 and 7 show the experimentally determined responses of the motor speed and the reference torque to step changes in the reference speed and the load torque. The noise of the constant motor speed is caused by the torque pulsations of the motor, which are caused by the permanent magnets in the magnetic circuit of the motor. The amplitude of this pulsating torque component is less than 0.05 Nm, but due to the low moment of inertia J, it manifests itself as a visible noise in the motor speed. The frequency of the pulsating component of the motor torque depends on the motor speed.

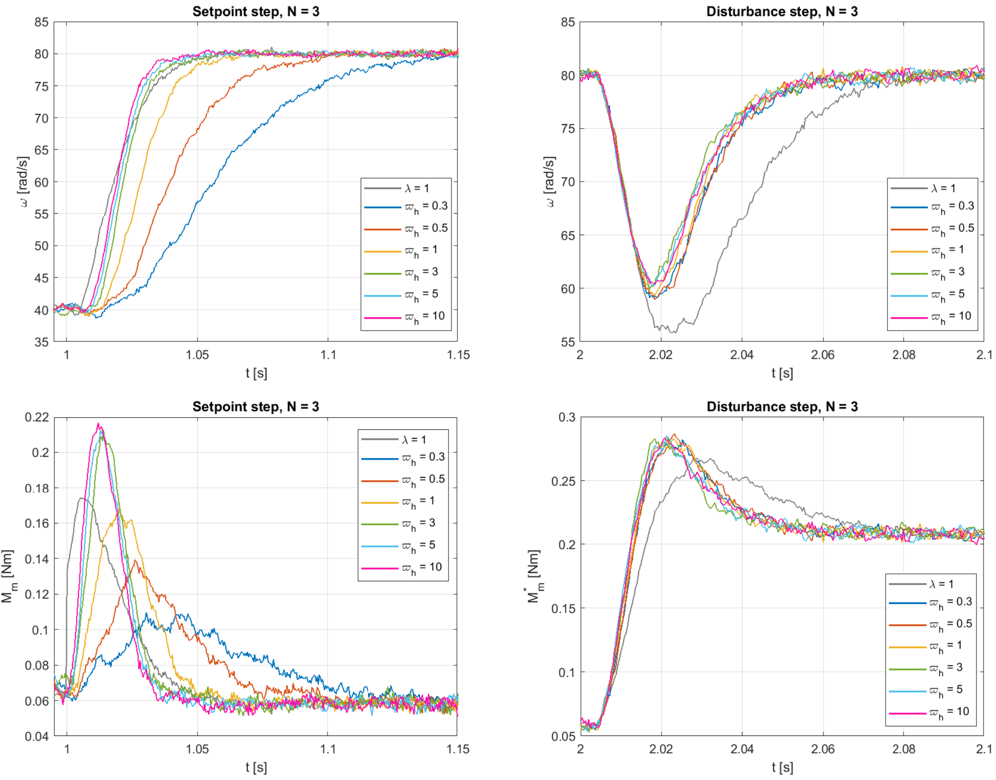

Figure 6. The responses of actual speed ω and reference torque M_m^* to reference speed and load torque steps with integer PI ($\lambda = 1$) and FOPI speed controller, for $N = 3$ and $\overline{\omega}_h = \text{var}$.

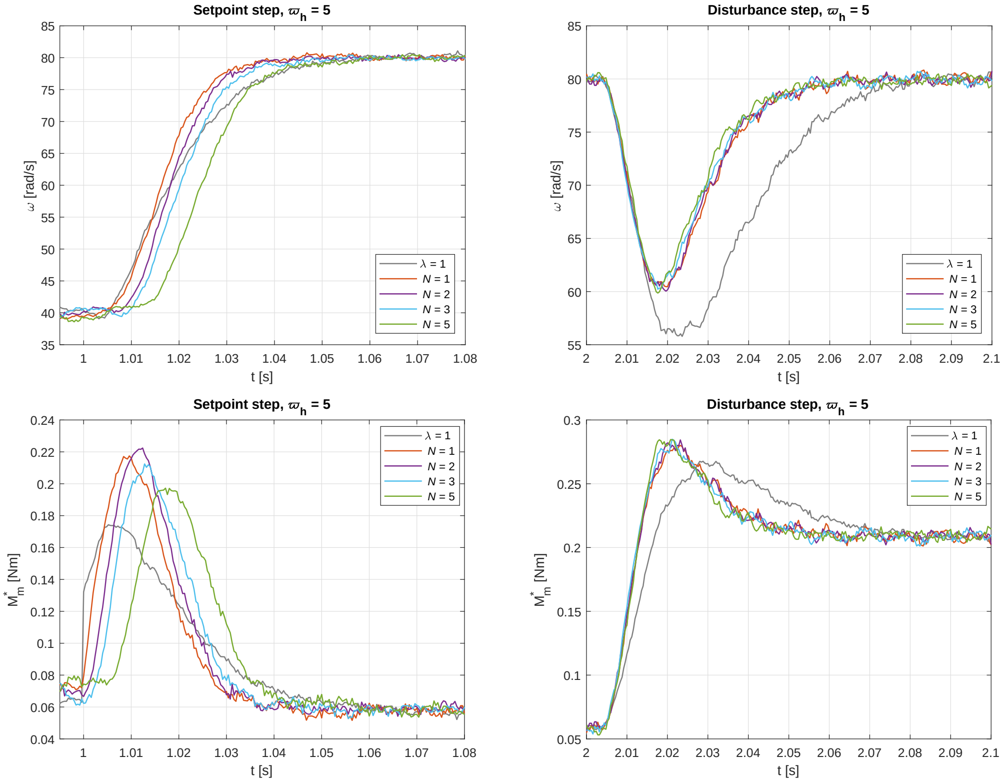

Figure 7. The responses of actual speed ω and reference torque M_m^* to reference speed and load torque steps with integer PI ($\lambda = 1$) and FOPI speed controller, for $\overline{\omega}_h = 5$ and $N = \text{var}$.

In the experiments, the results of which are shown in Figure 6, the fractional order integrator was approximated by the third order transfer function ($N = 3$) for different values of the upper limit of the normalized frequency band of the integrator $\overline{\omega}_h$. The reference tracking performance improves with increasing $\overline{\omega}_h$. In contrast, the disturbance rejection performance does not visibly depend on $\overline{\omega}_h$.

The results of experiments with the same upper limit of the normalized frequency band of the integrator ($\overline{\omega}_h = 5$), but with different order N ($N \in \{1,2,3,5\}$), are shown in Figure 7. The fastest tracking response is obtained with $N = 1$, i.e., with the simplest integrator approximation. Similar to the previous experiment, the disturbance rejection performance does not seem to depend on N.

4. Discussion

This section compares the expected IAE values of the normalized control loop with PI and FOPI controllers. It also evaluates the extent to which the experimental IAE values match the predicted values.

4.1. Comparison of Loop Dynamics with FOPI and PI Controllers for Normalized Control Loop

As already mentioned, Tables 3–13 show the normalized FOPI parameters that achieve optimum disturbance rejection with the permissible shape deviation of the control signal. In the mentioned tables, the calculated values of $\overline{\omega}_{b,opt}$, $\zeta_{0,opt}$, λ_{opt}, $\overline{K}_{p,opt}$, and $\overline{K}_{i,opt}$ for the specified upper limit of the integrator frequency band $\overline{\omega}_h$ and the specified order of the approximated integrator N are listed.

The values of $\overline{IAE}_r$ and $\overline{IAE}_d$ from Tables 3–13 versus $\overline{\omega}_h$ for $N = 1,2,3$ and 5 are shown in Figure 8. The horizontal gray lines represent the optimal value of $\overline{IAE}_r = 4.1214$ and $\overline{IAE}_d = 12.6387$ with an integer order PI controller ($\lambda = 1$).

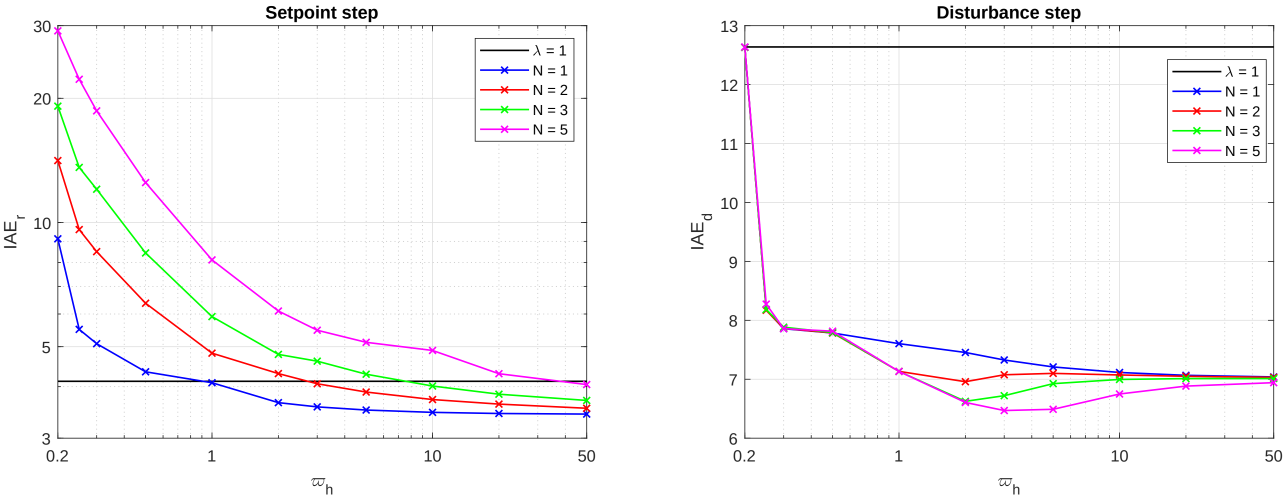

Figure 8. Plots of $\overline{IAE}_r$ and $\overline{IAE}_d$ versus $\overline{\omega}_h$ for $N = 1,2,3,5$ ($IAE_r = \overline{IAE}_r$, $IAE_d = \overline{IAE}_d$).

The comparison of the $\overline{IAE}_r$ and $\overline{IAE}_d$ waveforms for integer-order and fractional-order PI controllers in Figure 8 shows the following:

$$\begin{array}{lll} \overline{IAE}_r\big|_{\lambda=1} < \overline{IAE}_r\big|_{\lambda\neq 1} & \text{for} & \overline{\omega}_h < 1 \\ \overline{IAE}_d\big|_{\lambda=1} > \overline{IAE}_d\big|_{\lambda\neq 1} & \text{for} & \overline{\omega}_h \geq 0.2 \end{array} \tag{49}$$

The higher the order of the approximated integrator, the higher the $\overline{IAE}_r$. The lowest value of $\overline{IAE}_r$ is reached for $N = 1$.

The $\overline{IAE}_d$ values are always lower for $\overline{\omega}_h \in \langle 0.2, 50\rangle$ than when using an integer order PI controller. The lowest value $\overline{IAE}_d = 6.4903$ is obtained for $N = 5, \overline{\omega}_h = 5$. This value of $\overline{IAE}_d$ is 46.1% less than the value of $\overline{IAE}_d = 12.0387$ obtained with the integer order PI controller. The order of the integrator approximation N has no significant influence on the value of $\overline{IAE}_d$. The values of $\overline{IAE}_d$ for $N = 1$ and $N = 5$ differ from each other in a range of 0.014 to 13.26 percent. The smallest difference is at $\overline{\omega}_h = 0.3$ and the largest difference at $\overline{\omega}_h = 3$. It is to be expected that the $\overline{IAE}_d$ value would be even lower at around $\overline{\omega}_h = 3$ for $N > 5$.

4.2. Comparison of Expected and Actual Control Loop Characteristics

To check the correctness of the conversion of the normalized controller parameters given in Tables 3–13 to parameters for the real system, the predicted values of IAE_r and IAE_d were calculated for each experiment. These were compared with the actual values from the experiments in Table 16.

In Table 16, the IAE values for the setpoint and disturbance steps are denoted as $IAE_{r,exp}$ and $IAE_{d,exp}$, respectively. The values of $IAE_{r,exp}$ were calculated from the actual speed samples within the time interval from 1 s to 1.15 s. The values of $IAE_{d,exp}$ were calculated from the actual speed samples within the time interval from 2.005 s to 2.115 s. For comparison, the theoretical IAE values were calculated from

$$IAE_{r,opt} = \overline{IAE}_r T_d(\omega_2^* - \omega_1^*)\,, \quad IAE_{d,opt} = \overline{IAE}_d K_s T_d^2 (M_{L,2} - M_{L,1}) \tag{50}$$

and are also listed in Table 16. The variables δIAE_r and δIAE_d were calculated using the following formula

$$\delta IAE_r = 100 \times \frac{IAE_{r,exp} - IAE_{r,opt}}{IAE_{r,opt}}\,, \quad \delta IAE_d = 100 \times \frac{IAE_{d,exp} - IAE_{d,opt}}{IAE_{d,opt}} \tag{51}$$

and represent the relative error between the IAE values in the experiment and the IAE calculated from (50). The values in Table 16 show that the relative error between the calculated values of IAE and IAE from the experiment is less than four percent, indicating a high level of agreement between the model and the real system.

Table 16. Integrals of absolute error for setpoint and disturbance steps.

$\overline{\omega}_h$	N	$\mathbf{IAE_{r,exp}}$	$\mathbf{IAE_{r,opt}}$	$\boldsymbol{\delta}\mathbf{IAE_r}$ [%]	$\mathbf{IAE_{d,exp}}$	$\mathbf{IAE_{d,opt}}$	$\boldsymbol{\delta}\mathbf{IAE_d}$ [%]
–	–	0.84349	0.85725	−1.61	0.79857	0.78866	1.26
0.3	3	2.46165	2.50428	−1.70	0.49976	0.49196	1.59
0.5	3	1.73665	1.75469	−1.03	0.49411	0.48625	1.62
1	3	1.21870	1.23007	−0.92	0.45999	0.47451	−3.06
2	3	0.99412	0.99601	−0.19	0.42606	0.41344	3.05
3	3	0.94369	0.95129	−0.80	0.42277	0.41941	2.87
5	3	0.88048	0.89182	−1.27	0.44311	0.43215	2.54
10	3	0.82341	0.83491	−1.38	0.45167	0.43661	3.45
5	1	0.72675	0.73021	−0.47	0.46402	0.44985	3.15
5	2	0.79895	0.80733	−1.04	0.45325	0.44303	2.31
5	5	1.06061	1.06562	−0.47	0.41472	0.40500	2.40

As expected, the minimum value of $IAE_{d,exp} = 0.41472$ is at $\overline{\omega}_h = 5$, $N = 5$. This value is 48.1% less than the value of $IAE_{d,exp} = 0.79857$ obtained with the integer order PI controller.

4.3. Comparison with Results from Other Papers

The speed control loop of a servo drive with a torque generator and an FOPI speed controller is presented in [7,8,10,12]. The speed of a DC motor [7], a PMSM [7,8,10] and an induction motor [12] was controlled in these papers. In both the experimental servo drives [7,8,12] and the simulations [10], the dynamics of the torque generator response was conditioned by the tuning of the torque generator.

In our case, the torque generator channel exhibits a rather significant transport delay $T_{GM} = 5$ ms, which is due to the internal parameter settings of the ACOPOS P3 frequency inverter. We were not able to reduce this delay. It is clear that this transport delay affects the dynamics of the speed control loop, which is reflected in a slower response to the setpoint step and a higher control error at the load torque step than in a servo drive with a faster torque generator (or with a lower transport delay T_{GM}). The control error at the disturbance step is also influenced by the gain K_s, which in our case was several times higher than in the reported papers.

The presented speed controller tuning method was designed to achieve the fastest possible speed response. However, a first look at the simulation and experimental results presented in [7,8,10,12] shows a faster response and smaller control inaccuracy for the load torque step than in our paper. This is due to the high transport delay of our torque generator T_{GM} and the high value of K_s.

In order to roughly compare the characteristics of the speed control loop of the servo drive with FOPI controller tuned by the method presented in this paper with the results presented in cited papers [7,8,10,12], simulations were performed and the magnitude of the speed error $\Delta\omega = \omega^* - \omega$ and the settling time T_{set} were evaluated. The values of K_s and the value of the load torque step M_L were set in the simulations based on the data presented in the respective paper. The torque generator transport delay T_{GM} was not specified in any of the papers and was therefore set to the same value in all simulations, $T_{GM} = 0.5$ ms, which approximately corresponds to the real transport delays of the current control loops in the torque generators of the DC motors and the PMSMs. The FOPI controller parameters in the simulations were calculated for $\overline{\omega}_h = 3$, $N = 3$. The results of the simulations are listed in Table 17. The index "No" indicates results from the cited papers.

Table 17. Speed error and speed settling time in servo drives [7,8,10,12] compared to a servo drive tuned by the presented method.

No	K_s [kg^{-1}m^{-2}]	M_L [Nm]	$\Delta\omega_{No}$ [rad/s]	$T_{set,No}$ [ms]	$\Delta\omega$ [rad/s]	T_{set} [ms]
[7]	167	2.2	2	147	0.42	10
[8]	500	5	1	20	0.22	10
[10]	167	2	1	15	0.42	10
[12]	120	6	1.4	40	0.85	10

$T_{GM} = 0.5$ ms, $\overline{\omega}_h = 3$, $N = 3$.

The results in Table 17 show that the tuning of the FOPI controller parameters by the presented method achieves lower speed error and shorter settling time for the load torque step than those obtained in the cited papers.

5. Conclusions

This article describes a method for tuning the parameters of a fractional-order PI (FOPI) speed controller. The tuning of the controller parameters is based on minimizing the value of IAE at a disturbance step, while limiting the deviation of the control signal from an ideal 1P pulse. These two requirements make it possible to minimize the settling time with an overshoot of zero or almost zero.

The tuning of the controller parameters is performed for a normalized process (delay and gain equal to 1) and a controller where the fractional order integrator is approximated by a rational transfer function. The approximation of the fractional order integrator is based on the Oustaloup method and guarantees a zero error in the steady state under disturbance. Tables with normalized parameters of the FOPI controller were calculated for eleven values of the upper limit of the frequency band and for four orders ($N = 1, 2, 3, 5$) of the integrator approximated by the Oustaloup method. The tables also contain the values of the integral of the absolute error (IAE) at the setpoint and disturbance steps for the normalized process.

The characteristics of the control loop with the FOPI controller were compared with those of the integer order PI controller. The aim was to compare the values of the IAE at the setpoint and disturbance steps. The parameters of the integer order PI controller were set to minimize the IAE at the disturbance step. The comparison of the IAE values for an integer order PI controller and a FOPI controller confirms that the FOPI controller can significantly increase the tracking and disturbance rejection performance. In the best case, the IAE value in the disturbance step of the FOPI controller is almost half of the IAE value with an ordinary PI controller (integer order).

When using a fractional-order PI controller in a real servo drive, the actual controller parameters are recalculated from the normalized parameters based on the actual characteristics of the servo drive. Similarly, the actual IAE values can be calculated from the tabulated values.

The characteristics of the control loop were tested on a servo system with a pair of industrial servo drives. The responses of the motor speed to setpoint speed and load torque steps were evaluated. The IAE values at the setpoint and disturbance steps were calculated from the actual rotor speed samples. Comparison of the IAE values calculated from the experimental results and the tabulated values for the normalized system confirmed the accuracy of the controller design and the correctness of the calculated IAE values for the normalized control loop.

Finally, the limitations of this study can be briefly summarized as follows. First, the optimized parameters of the normalized FOPI controller presented in Section 3.1 are only applicable when controlling an IPDT system. An electric servo drive with a torque generator can be considered as such a system. Second, the fractional-order integrator in the FOPI controller must be approximated by the Oustaloup method, which results in a continuous transfer function of the chosen order. A proper conversion method to discrete form must be applied in the digital implementation. Third, the optimized FOPI controller parameters were computed for the first to fifth order Oustaloup approximation of the fractional order integrator. The computations were not performed for higher orders due to the increased computational requirements on the computer executing the FOPI controller parameter optimization algorithm. Fourth, the optimized FOPI controller parameters were not calculated even for $\overline{\omega}_h > 50$. However, from the plots of $\overline{IAE}_r$ and $\overline{IAE}_d$ versus $\overline{\omega}_h$, it can be inferred that higher values of $\overline{\omega}_h$ would not result in a significant reduction in the values of $\overline{IAE}_r$ and $\overline{IAE}_d$. Fifth, the application and applicability of the calculated parameters of the normalized FOPI controller were experimentally validated on the servo drive, which allowed to correctly verify the behavior of the speed control loop only for $\overline{\omega}_h \leq 5$.

Author Contributions: Writing—original draft preparation, I.B., I.B.J., P.B., M.H. and D.V.; writing—review and editing, I.B., P.B., M.H. and D.V.; simulations, I.B.; experiments, I.B.J.; project administration, P.B. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported in part by the following grants: Grant No. 1/0637/23 financed by the Scientific Grant Agency of the Ministry of Education, Research, Development and Youth of the Slovak Republic (VEGA); Grant No. APVV-21-0125 financed by the Slovak Research and Development Agency; Research Program P2-0001 (Systems and Control) and research project L2-3166 (Supervisory control system for plant-wide optimization of wastewater treatment plant operation) financed by the Slovenian Research and Innovation Agency.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Data are contained within the article.

Acknowledgments: Supported by E-Academia Slovaca, a non-profit organisation, Sadmelijská 1, 831 06 Bratislava, Slovakia.

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations

The following abbreviations are used in this manuscript:

DC	Direct Current
FOPI	Fractional-Order Proportional-Integral
FOPID	Fractional-Order Proportional-Integral-Derivative
IAE	Integral of Absolute Error
IE	Integral of Error
IPDT	Integrator Plus Dead-Time
LESO	Linear Extended State Observer
MDPM	Multiple Dominant Pole Method
OPC UA	Open Platform Communications Unified Architecture
PI	Proportional-Integral
PID	Proportional-Integral-Derivative
PMSM	Permanent Magnet Synchronous Motor
PPM	Performance Portrait Method
TV	Total Variance

Nomenclature

The following symbols are used in this manuscript:

k	Cycle number of the FOPI controller parameter optimization algorithm
k_d	Gain of the derivative component of the FOPID controller
k_i	Gain of the integral component of the FOPID controller
k_{max}	The total number of cycles of the FOPI controller parameter optimization algorithm
k_p	Gain of the proportional component of the FOPID controller
n_N	Rated speed of the motor
s	Laplace operator
s_0	Absolute value of the double real pole in the speed control loop transfer function
t	Time
t_1	Time of the reference speed step in the experiment
t_2	Time of the load torque step in the experiment
E	Control error
$E_{\infty,A}$	Steady-state error at the setpoint step
$E_{\infty L,A}$	Steady-state error at the disturbance step
$F_I(s)$	Transfer function of the reference speed filter
$F_I(\xi)$	Transfer function of the normalized reference speed filter
$G_{1,d}(\xi)$	Closed loop transfer function for a disturbance step with the normalized PI controller
$G_{1,r}(\xi)$	Closed loop transfer function with the normalized PI controller
$G_{e,A}(s)$	Error transfer function for a setpoint step
$G_{eL,A}(s)$	Error transfer function for a disturbance step
$G_d(s)$	Closed loop transfer function for a disturbance step with the FOPI controller
$G_d(\xi)$	Normalized $G_d(s)$
$\widehat{G}_o^{\kappa}(s)$	Transfer function of the fractional-order derivative approximated by the Oustaloup method
$G_r(s)$	Closed loop transfer function with the FOPI controller
$G_r(\xi)$	Normalized $G_r(s)$
$\mathbf{H}^k$	Three-dimensional space of parameters $\overline{\omega}_b$, ξ_0, and λ in the k-th cycle of the optimization algorithm
$IAE_{d,exp}$	The value of the integral of absolute error at the disturbance step retrieved from the experiment
$IAE_{d,opt}$	Calculated value of the integral of absolute error at the disturbance step when using the optimized parameters of the normalized FOPI controller
$IAE_{r,exp}$	The value of the integral of absolute error at the setpoint step retrieved from the experiment
$IAE_{r,opt}$	Calculated value of the integral of the absolute error at the setpoint step when using the optimized parameters of the normalized FOPI controller
$\overline{IAE}_d$	Normalized integral of absolute error at the disturbance step

$\overline{IAE}_{d,min}$	Minimum $\overline{IAE}_d$ value
$\overline{IAE}_{d,opt}$	Integral of absolute error at the disturbance step when the optimized parameters of the normalized FOPI controller are used
$\overline{IAE}_{r,min}$	Minimum $\overline{IAE}_r$ value
$\overline{IAE}_{r,opt}$	Integral of absolute error at the setpoint step when the optimized parameters of the normalized FOPI controller are used
IE_d	Integral of error at the disturbance step
$\overline{IE}_d$	Normalized integral of error at the disturbance step
$\overline{IE}_{d,min}$	Minimum $\overline{IE}_d$ value
IE_r	Integral of error at the setpoint step
$\overline{IE}_r$	Normalized integral of absolute error at the setpoint step
$\overline{IE}_{r,min}$	Minimum $\overline{IE}_r$ value
J	Moment of inertia
K_i	Gain of the integral component of the FOPI controller
$\overline{K}_i$	Normalized K_i
$\overline{K}_{i,opt}$	The value of $\overline{K}_i$ after the optimization algorithm has been completed
K_o	Coefficient in the modified form of the Oustaloup approximation
$\overline{K}_o$	Normalized K_o
K_p	Gain of the proportional component of the FOPI controller
$\overline{K}_p$	Normalized K_p
$\overline{K}_{p,opt}$	The value of $\overline{K}_p$ after the optimization algorithm has been completed
K_s	Gain of the system
$M_A(s)$	Numerator polynomial of a transfer function approximating a fractional-order derivative or integral
$M_I(s)$	Numerator polynomial in the modified form of the Oustaloup approximation
$M_I(\xi)$	Numerator polynomial in the normalized Oustaloup approximation
M_L	Load torque
$\overline{M}_L$	Normalized load torque
M_L^*	Reference load torque
M_m	Motor torque
$\overline{M}_m$	Normalized motor torque
M_m^*	Reference motor torque
M_{m1}^*	Reference torque of the motor No. 1
M_{m2}^*	Reference torque of the motor No. 2
$M_{Od}(s)$	Numerator polynomial of the transfer function $G_d(s)$
$M_{Od}(\xi)$	Numerator polynomial of the transfer function $G_d(\xi)$
$M_{Or}(s)$	Numerator polynomial of the transfer function $G_r(s)$
$M_{Or}(\xi)$	Numerator polynomial of the transfer function $G_r(\xi)$
N	Order of the transfer function of a fractional-order integrator approximated by the Oustaloup method
$N_A(s)$	Denominator polynomial of a transfer function approximating a fractional-order derivative or integral
$N_I(s)$	Denominator polynomial in the modified form of the Oustaloup approximation
$N_I(\xi)$	Denominator polynomial in the normalized Oustaloup approximation
$N_O(s)$	Denominator polynomial of the transfer functions $G_d(s)$ and $G_r(s)$
$N_O(\xi)$	Denominator polynomial of the transfer functions $G_d(\xi)$ and $G_r(\xi)$
N_{oP}	The number of all tested values of a particular variable (i.e., the values of $\overline{\omega}_b$, ξ_0, or λ) applied in one cycle of the optimization algorithm
P_n	Designates one of the parameters: $\overline{\omega}_b$, ξ_0, or λ
$P_{n,max}^k$	Maximum value of parameter P_n in k-th cycle of optimizing algorithm
$P_{n,min}^k$	Minimum value of parameter P_n in k-th cycle of optimizing algorithm
$P_{n,opt}^{k-1}$	Optimum value of parameter P_n in the optimizing algorithm cycle $k-1$
P_N	Rated power of the motor
T_d	Transport delay in the control channel of the system
T_{GM}	Transport delay of the torque generator
T_s	Sampling period of the discrete speed controller
T_{set}	Settling time
$TV_1(\overline{U}^*)$	Total variation in the normalized control signal from the one-pulse function

$TV_{1d}(\overline{U}^*)$	Total variation in the normalized control signal from the one-pulse function at the disturbance step
$TV_{1r}(\overline{U}^*)$	Total variation in the normalized control signal from the one-pulse function at the setpoint step
U	The output of the FOPI controller
$\overline{U}^*$	Normalized control signal
$\overline{U}^*_0$	Initial value of the normalized control signal
$\overline{U}^*_c$	Sample No. c of the normalized control signal
$\overline{U}^*_{max}$	Maximum value of the normalized control signal
$\overline{U}^*_\infty$	Steady state value of the normalized control signal
${}_aD_t^{\mu}$	Integrodifferentiator operator, where a, t are the limits of the operation
γ_j	Angular frequency representing a pole in the Oustaloup approximation
γ'_j	Angular frequency representing a zero in the Oustaloup approximation
δIAE_d	Relative deviation between experimental and calculated IAE_d value
δIAE_r	Relative deviation between experimental and calculated IAE_r value
ε_r	maximum allowable shape deviation at the setpoint or disturbance step
λ	Fractional order of the integral
λ^1_{max}	Maximum value of λ in the first cycle of the optimization algorithm
λ^1_{min}	Minimum value of λ in the first cycle of the optimization algorithm
λ_{opt}	The value of λ after the optimization algorithm has been completed
ξ	Normalized Laplace operator
ξ_0	Absolute value of the double dominant real pole in the normalized closed-loop transfer function
$\xi^1_{0,max}$	Maximum value of ξ_0 in the first cycle of the optimization algorithm
$\xi^1_{0,min}$	Minimum value of ξ_0 in the first cycle of the optimization algorithm
$\xi_{0,opt}$	The value of ξ_0 after the optimization algorithm has been completed
μ	Fractional order of the derivative
κ	Fractional order of the derivative
φ_1	Actual angular position of motor M_1
φ_2	Actual angular position of motor M_2
ω	Actual angular speed
ω^*	Reference angular speed
ω^*_1, ω^*_2	The reference values of the motor speed in the experiment
ω_b	The lower limit of the frequency band in which the fractional-order derivative/integrator approximation is valid when the Oustaloup method is used
$\overline{\omega}_b$	Normalized ω_b
$\overline{\omega}^1_{b,max}$	Maximum value of $\overline{\omega}_b$ in the first cycle of the optimization algorithm
$\overline{\omega}^1_{b,min}$	Minimum value of $\overline{\omega}_b$ in the first cycle of the optimization algorithm
$\overline{\omega}_{b,opt}$	The value of $\overline{\omega}_b$ after the optimization algorithm has been completed
ω_h	Upper limit of the frequency band in which the fractional-order derivative/integrator approximation is valid when the Oustaloup method is used
$\overline{\omega}_h$	Normalized ω_h
ω_j	Angular frequency representing a pole in the modified form of the Oustaloup approximation
$\overline{\omega}_j$	Normalized ω_j
ω'_j	Angular frequency representing a zero in the modified form of the Oustaloup approximation
$\overline{\omega}'_j$	Normalized ω'_j
Γ_d	Optimizing function used to tune $\overline{\omega}_b$, ξ_0, and λ values
ΔP^k_n	Step to change the value of one of the parameters $\overline{\omega}_b$, ξ_0, or λ in the k-th cycle of the optimization algorithm
$\Delta\omega$	Speed error at load torque step
Λ^k	Interval of λ values to be applied in the k-th cycle of the optimization algorithm
Ξ^k_0	Interval of ξ_0 values to be applied in the k-th cycle of the optimization algorithm
Ω^k_b	Interval of ω_b values to be applied in the k-th cycle of the optimization algorithm
$\mathcal{L}$	Laplace transform

References

1. Wiora, J.; Wiora, A. Influence of Methods Approximating Fractional-Order Differentiation on the Output Signal Illustrated by Three Variants of Oustaloup Filter. *Symmetry* **2020**, *12*, 1898. [CrossRef]
2. Vinagre, B.; Podlubný, I.; Hernández, A.; Feliu, V. Some approximations of fractional order operators used in control theory and applications. *J. Fract. Calc. Appl. Anal.* **2000**, *4*, 47–66.
3. Shah, P.; Agashe, S. Review of fractional PID controller. *Mechatronics* **2016**, *38*, 29–41. [CrossRef]
4. Diethelm, K.; Ford, N.; Luchko, Y. Algorithms for the fractional calculus: A selection of numerical methods. *Comput. Methods Appl. Mech. Eng.* **2005**, *194*, 743–773. [CrossRef]
5. Podlubný, I. Fractional-Order Systems and PI$^{\lambda}$D$^{\mu}$-Controllers. *IEEE Trans. Autom. Control.* **1999**, *44*, 208–214. [CrossRef]
6. Petráš, I. Fractional-Order feedback control of a DC motor. *J. Electr. Eng.* **2009**, *60*, 117–128.
7. Lino, P.; Maione, G.; Stasi, S.; Padula, F.; Visioli, A. Synthesis of Fractional-order PI Controllers and Fractional-order Filters for Industrial Electrical Drives. *IIEEE/CAA J. Autom. Sin.* **2017**, *4*, 58–69. [CrossRef]
8. Chen, L.; Chen, G.; Li, P.; Lopes, A.; Tenreiro Machado, J.; Xu, S. A variable-order fractional proportional-integral controller and its application to a permanent magnet synchronous motor. *Alex. Eng. J.* **2020**, *59*, 3247–3254. [CrossRef]
9. Jakovljević, B.; Lino, P.; Maione, G. Optimized Current and Speed Fractional-Order PID Control in Electrical Drives. In *Proceedings of the International Conference on Fractional Differentiation and Its Applications (ICFDA'21)*; Dzielinski, A., Sierociuk, D., Ostalczyk, P., Eds.; Springer: Cham, Switzerland, 2022; pp. 141–152. [CrossRef]
10. Apte, A.; Thakar, U.; Joshi, V. Disturbance observer based speed control of PMSM using fractional order PI controller. *IEEE/CAA J. Autom. Sin.* **2019**, *6*, 316–326. [CrossRef]
11. Wang, S.; Li, B.; Chen, P.; Yu, W.; Peng, Y.; Luo, Y. A fractional-order active disturbance rejection control for permanent magnet synchronous motor position servo system. *Asian J. Control* **2024**, *26*, 3137–3147. [CrossRef]
12. Kumar, D.M.; Mudaliar, H.K.; Cirrincione, M.; Mehta, U.; Pucci, M. Design of a Fractional Order PI (FOPI) for the Speed Control of a High-Performance Electrical Drive with an Induction Motor. In Proceedings of the 21st International Conference on Electrical Machines and Systems (ICEMS), Jeju, Republic of Korea, 7–10 October 2018; pp. 1198–1202. [CrossRef]
13. Angel, L.; Viola, J. Control Performance Assessment of Fractional-Order PID Controllers Applied to Tracking Trajectory Control of Robotic Systems. *Wseas Trans. Syst. Control* **2022**, *17*, 62–73. [CrossRef]
14. Bélai, I.; Huba, M.; Burn, K.; Cox, C. PID and filtered PID control design with application to a positional servo drive. *Kybernetika* **2019**, *55*, 540–560. [CrossRef]
15. Phung Quang, N.; Dittrich, J.A. *Vector Control of Three-Phase AC Machines—System Development in the Practice*, 2nd ed.; Springer: Berlin/Heidelberg, Germany, 2015. [CrossRef]
16. Oustaloup, A.; Levron, F.; Mathieu, B.; Nanot, F. Frequency-band complex noninteger differentiator: Characterization and synthesis. *IEEE Trans. Circuits Syst. I Fundam. Theory Appl.* **2000**, *47*, 25–39. [CrossRef]
17. Deniz, F.; Alagoz, B.; Tan, N.; Koseoglu, M. Revisiting four approximation methods for fractional order transfer function implementations: Stability preservation, time and frequency response matching analyses. *Annu. Rev. Control* **2020**, *49*, 239–257. [CrossRef]
18. Singhal, R.; Padhee, S.; Kaur, G. Design of Fractional Order PID Controller for Speed Control of DC Motor. *Int. J. Sci. Res. Res. Publ.* **2012**, *2*, 1–8.
19. Leuzzi, R.; Lino, P.; Maione, G.; Stasi, S.; Padula, F.; Visioli, A. Combined Fractional Feedback-Feedforward Controller Design for Electrical Drives. In Proceedings of the ICFDA'14 International Conference on Fractional Differentiation and Its Applications 2014, Catania, Italy, 23–25 June 2014; pp. 1–6. [CrossRef]
20. Dulf, É.H.; Şuşcă, M.; Kovács, L. Novel Optimum Magnitude Based Fractional Order Controller Design Method. *IFAC-PapersOnLine* **2018**, *51*, 912–917. [CrossRef]
21. Wang, R.; Pi, Y. Fractional-order PI Speed Control for Permanent Magnet Synchronous Motor. In Proceedings of the 2012 IEEE International Conference on Mechatronics and Automation, Chengdu, China, 5–8 August 2012; pp. 2303–2308. [CrossRef]
22. Li, X.; Gao, L. A Simple Frequency-domain Tuning Method of Fractional-order PID Controllers for Fractional-order Delay Systems. *Int. J. Control Autom. Syst.* **2022**, *20*, 2159–2168. [CrossRef]
23. Pradhan, R.; Majhi, S.K.; Pradhan, J.K.; Pati, B.B. Optimal fractional order PID controller design using Ant lion Optimizer. *Ain Shams Eng. J.* **2020**, *11*, 281–291. [CrossRef]
24. Dastjerdi, A.A.; Vinagre, B.M.; Chen, Y.; HosseinNia, S.H. Linear fractional order controllers; A survey in the frequency domain. *Annu. Rev. Control* **2019**, *47*, 51–70. [CrossRef]
25. Verma, S.K.; Yadav, S.; Nagar, S.K. Optimization of Fractional Order PID Controller Using Grey Wolf Optimizer. *J. Control Autom. Electr. Syst.* **2017**, *28*, 314–322. [CrossRef]
26. Paducel, I.; Safirescu, C.O.; Dulf, E.H. Fractional Order Controller for Wind Turbines. *Appl. Sci.* **2022**, *12*, 8400. [CrossRef]
27. Bélai, I.; Bélai, I. Electric Servo Drive with Fractional-Order PI Speed Controller. In Proceedings of the 2023 24th International Conference on Process Control (PC), Strbske Pleso, Slovakia, 6–9 June 2023; pp. 6–11. [CrossRef]
28. Viteckova, M.; Vitecek, A. Use of multiple dominant pole method for controller tuning. In Proceedings of the 13th International Carpathian Control Conference (ICCC), High Tatras, Slovakia, 28–31 May 2012; pp. 757–762. [CrossRef]
29. Huba, M.; Skogestad, S.; Fikar, M.; Hovd, M.; Johansen, T.; Rohal'-Ilkiv, B. *Selected Topics on Constrained and Nonlinear Control*; STU Bratislava—NTNU Trondheim: Bratislava, Slovakia, 2011; p. 104.

30. Huba, M.; Bistak, P.; Vrancic, D. Parametrization and Optimal Tuning of Constrained Series PIDA Controller for IPDT Models. *Mathematics* **2023**, *11*, 4229. [CrossRef]
31. Huba, M. Performance measures, performance limits and optimal PI control for the IPDT plant. *J. Process. Control* **2013**, *23*, 500–515. [CrossRef]
32. Franklin, G.F.; Powell, J.D.; Workman, M. *Digital Control of Dynamic Systems*, 3rd ed.; Ellis-Kagle Press: Half Moon Bay, CA, USA, 2022.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.