mathematics

MDPI

Article

High-Order Filtered PID Controller Tuning Based on Magnitude Optimum

Damir Vrančić [1,*] and Mikuláš Huba [2]

[1] Jožef Stefan Institute, SI-1000 Ljubljana, Slovenia
[2] Faculty of Electrical Engineering and Information Technology, Slovak University of Technology in Bratislava, Ilovičova 3, 81219 Bratislava, Slovakia; mikulas.huba@stuba.sk
[*] Correspondence: damir.vrancic@ijs.si

Abstract: The paper presents a tuning method for PID controllers with higher-order derivatives and higher-order controller filters (HO-PID), where the controller and filter orders can be arbitrarily chosen by the user. The controller and filter parameters are tuned according to the magnitude optimum criteria and the specified noise gain of the controller. The advantages of the proposed approach are twofold. First, all parameters can be obtained from the process transfer function or from the measured input and output time responses of the process as the steady-state changes. Second, the a priori defined controller noise gain limits the amount of HO-PID output noise. Therefore, the method can be successfully applied in practice. The work shows that the HO-PID controllers can significantly improve the control performance of various process models compared to the standard PID controllers. Of course, the increased efficiency is limited by the selected noise gain. The proposed tuning method is illustrated on several process models and compared with two other tuning methods for higher-order controllers.

Keywords: higher-order controllers; PID controller; magnitude optimum; controller tuning; noise attenuation

check for updates

Citation: Vrančić, D.; Huba, M. High-Order Filtered PID Controller Tuning Based on Magnitude Optimum. *Mathematics* **2021**, *9*, 1340. https://doi.org/10.3390/math9121340

Academic Editor: Ioannis Dassios

Received: 6 May 2021
Accepted: 5 June 2021
Published: 9 June 2021

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Copyright: © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

The PID controllers are widespread in many industries and are frequently included in embedded solutions [1–4]. This is not surprising, since the basic PID control algorithm is very simple and the control performance, when the controller is tuned appropriately, is usually very good. However, the control performance can be improved by increasing the controller order. The improvement depends on the process order. While the first-order process can be efficiently controlled by the PI controller and the second-order process by the PID controller, the control efficiency for higher-order processes can be improved by increasing the controller order beyond the PID control.

In practice the PI controllers are used more often than the PID controllers, since the latter significantly increase the controller output noise. Naturally, with higher degrees of controllers, the problem becomes aggravated. Therefore, the appropriate higher-order controller filter is inevitable in practical applications.

For easier classification of the HO-PID controllers according to the controller (m) and filter (n) order, let us denote them as $\mathrm{PID}_\mathrm{n}^\mathrm{m}$. A general $\mathrm{PID}_\mathrm{n}^\mathrm{m}$ controller transfer function $G_{CF}(s)$ can be defined as follows:

$$\begin{gathered} G_{CF}(s) = G_C(s)G_F(s), \quad where \\ G_C(s) = \left(K_{-1}s^{-1} + K_0 + K_1 s + \cdots + K_m s^m\right) \\ G_F(s) = \tfrac{1}{(1+T_F s)^n} \end{gathered} \tag{1}$$

where K_{-1}, K_0, K_1, … K_m are controller gains, and $G_F(s)$ is the binomial filter with filter time constant T_F. In practical applications, in order to limit the higher-frequency controller output noise, $n \geq m$. Note that PID_0^0 denotes the PI controller ($K_I = K_{-1}$, $K_P = K_0$) and PID_1^1 denotes the PID controller ($K_I = K_{-1}$, $K_P = K_0$, $K_D = K_1$) with the first-order filter ($n = 1$).

Several tuning methods for HO-PID controllers have been proposed so far. The majority of them are made for proportional-integrative-derivative-accelerative (PIDA) controllers ($\mathrm{PID}_\mathrm{n}^2$). The controller structure is either 1 degree of freedom (1-DOF) [5–20] or 2 degrees of freedom (2-DOF) [21–23] which can optimise the tracking and control performance.

The tuning methods for the mentioned PIDA controllers are derived either for the first-order process with delay [20,22,23], third order process [5,11,13,14,21], first-order double integrating process [5,11,16], second-order integrating process [8,9,11,16], double integrating process with time delay [10], fourth-order system [18], for different types of process models [6,17,18] or for the automatic voltage regulator (AVR) in the generator excitation system [7,15,19]. Unfortunately, only a few of the mentioned PIDA controller tuning methods take into account the controller filter in the controller design stage [5–7,10,15,23]. Therefore, the practical implementation of other PIDA tuning methods remains questionable.

Besides PIDA controllers, some higher-order controller tuning methods also exist [24–27]. The tuning method for the PID_0^3 controller (the controller filter is not considered), where the controlled process is a model of the ship power plant, including the heat exchanger, is given in [24]. The method optimises the IAE for disturbance rejection while limiting the peak of the closed-loop amplitude frequency response.

Tuning methods for even higher-order controllers ($m > 3$) were developed for the integrating process model with a time delay (IPTD) [25–27]. Although the type of the process model seems to be limited, we have to mention that many stable process models can be modelled as IPTD processes [1]. For HO-PID control of stable time-delayed processes, a new method was also proposed by generalizing Skogestad's method SIMC [28]. The basic version is based on the approximation of processes by transfer functions with multiple time constants (obtained, for example, by an appropriate identification method); however, a suitable model can also be obtained from a more general description of the process reduced by the modified "half-rule" method [29]. Although not specifically designed for HO-PID controllers, we should also mention that the tuning approach is based on the design of multiple dominant closed-loop poles for delayed processes, applied to the PI and PID controllers [30], which can be easily extended to HO-PID controllers.

The developed tuning methods for the $\mathrm{PID}_\mathrm{n}^\mathrm{m}$ controllers reveal that the HO-PID controllers can be much more efficient than the ordinary PID controllers without significant increase of the controller output noise.

This paper presents the $\mathrm{PID}_\mathrm{n}^\mathrm{m}$ controller and filter tuning method, which is based on the parametric or the non-parametric process description. It means that the process can be given by the general transfer function (of the arbitrary order and time delay) or by the process input and output time-responses during the steady-state change of the process. The only user-defined parameters will be the controller (m) and the filter (n) order and the desired high-frequency gain of the controller. As will be shown later, the controller parameters will be calculated analytically.

Therefore, the main advantages of the proposed method are the flexibility of the process description (the process model is not required), simple specifications by the user and simple calculation of the controller and filter parameters.

The content of the paper is as follows. The tuning method for the $\mathrm{PID}_\mathrm{n}^\mathrm{m}$ controllers is covered in Section 2. The calculation of the controller and controller filter time constant, according to the desired closed-loop high-frequency gain, is derived in Section 3. The comparison with some other tuning methods is carried out in Section 4. The paper concludes with Section 5.

2. HO-PID Controller Tuning

The HO-PID controller parameters will be derived according to the magnitude optimum multiple integration (MOMI) tuning method, which is based on the magnitude optimum (MO) criteria [31–37]. The main advantages of the MOMI method are that it combines frequency-domain MO tuning criterion (providing a fast and non-oscillatory closed-loop process output response) with the time-domain method of moments (the calculation of the process characteristic areas directly from the process time responses).

The process

The general order process transfer function with time delay is defined by the following expression:

$$G_P(s) = \frac{K_{PR}\left(1 + b_1 s + b_2 s^2 + \cdots + b_r s^r\right)}{1 + a_1 s + a_2 s^2 + \cdots + a_p s^p} e^{-sT_{del}} \tag{2}$$

where K_{PR} is the process gain, T_{del} is the process time delay and a_1 to a_p and b_1 to b_r are the process dynamic parameters. To simplify the derivation, let us assume that the process transfer function is developed into an infinite Taylor series around $s = 0$:

$$G_P(s) = G_{P0} + G_{P1}s + \frac{G_{P2}}{2!}s^2 + \frac{G_{P3}}{3!}s^3 + \cdots, \tag{3}$$

where G_{Pk} are the k-th derivatives of the $G_P(s)$ over s around $s = 0$. The moments can be calculated from the process impulse response $h(t)$ in the following way [1,38]:

$$G_P^{(k)}(0) = G_{Pk} = (-1)^k \int_0^\infty t^k h(t) dt \tag{4}$$

Besides measuring the process impulse response, the moments can also be calculated from the process steady-state change by measuring the process input and output time responses [32,34]. By integrating the process input and output time responses, the so-called characteristic areas A_k are obtained, which are related to the process moments as follows:

$$A_k = \frac{(-1)^k}{k!} G_{Pk} \tag{5}$$

The process transfer function, based on the characteristic areas, can be derived from (3) and (5) as follows:

$$G_P(s) = A_0 - A_1 s + A_2 s^2 - A_3 s^3 + \cdots. \tag{6}$$

Since the calculation of the mentioned areas from the process input and output time responses, during arbitrary steady-state change, are already covered in detail in [36], it will not be repeated herein.

The process moments (4) and, therefore, the characteristic areas A_k (5), can also be calculated from the process transfer function (2) by calculating the derivatives of $G_P(s)$ over s around $s = 0$. The result is the following [32,34]:

$$\begin{gathered}
A_0 = K_{PR} \\
A_1 = K_{PR}(a_1 - b_1 + T_{del}) \\
A_2 = A_1 a_1 + K_{PR}\left(b_2 - a_2 - T_{del} b_1 + \tfrac{T_{del}^2}{2!}\right) \\
\vdots \\
A_k = \sum_{i=1}^{k-1} (-1)^{k+i-1} A_1 a_{k-i} + (-1)^{k+1} K_{PR}(a_k - b_k) + K_{PR} \sum_{i=1}^{k} \tfrac{(-1)^{k+i}}{i!} T_{del}^i b_{k-i}
\end{gathered} \tag{7}$$

Therefore, the characteristic areas in expression (6) can be calculated either from the process time response or from the process transfer function. This is a very important advantage, since the actual process model can be used, but is not required.

In order to simplify the derivation of the controller parameters, the controller binomial filter $G_F(s)$ (1) will be considered as a part of the process. Since the above areas are calculated for the process without the filter, the areas A_i should be modified, accordingly. If the filter

$G_F(s)$ (1) is added to the process (2) and developed into a Taylor series, it can be derived that the new areas, denoted by A_{iF}, can be simply calculated as:

$$A_{VF} = M_F^n A_V,\ where$$
$$M_F = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ T_F & 1 & 0 & 0 & 0 & 0 \\ T_F^2 & T_F & 1 & 0 & 0 & 0 \\ T_F^3 & T_F^2 & T_F & 1 & 0 & 0 \\ T_F^4 & T_F^3 & T_F^2 & T_F & 1 & 0 \\ \vdots & \vdots & \vdots & \vdots & T_F & \ddots \end{bmatrix},\ A_V = \begin{bmatrix} A_0 \\ A_1 \\ A_2 \\ A_3 \\ A_4 \\ \vdots \end{bmatrix},\ A_{VF} = \begin{bmatrix} A_{0F} \\ A_{1F} \\ A_{2F} \\ A_{3F} \\ A_{4F} \\ \vdots \end{bmatrix} \tag{8}$$

Note that n is the binomial filter order (1). Naturally, the chosen size of the matrix and the vectors depends on the number of the required areas.

Note that the characteristic areas with the included controller filter can be obtained a-posteriori, when the process areas A_i are already measured either from the process time response (5) or calculated from the process transfer function (7).

For further reference, please note that the process areas with the included controller binomial filter are denoted with index F (A_{iF}) and the process areas without the controller filter are denoted without index F (A_i).

The closed-loop transfer function

In the paper, the process and the HO-PID controller (1) will be considered, as shown in Figure 1. Signals r, r_f, e, u, d, n and y stand for the reference, filtered reference, control error, controller output, process input disturbance, process output noise and the process output, respectively. Block G_{FR} represents the second order filter for the reference signal in order to reduce excessive controller output change on reference changes:

$$G_{FR}(s) = \frac{1}{(1 + T_{FR}s)^2}, \tag{9}$$

where T_{FR} denotes the reference filter time constant. Due to simplicity, the filter order in (9) is fixed. However, note that the filter order may be increased by increasing the controller order so as to additionally attenuate the swings of the signal u for step-like changes of the reference signal r.

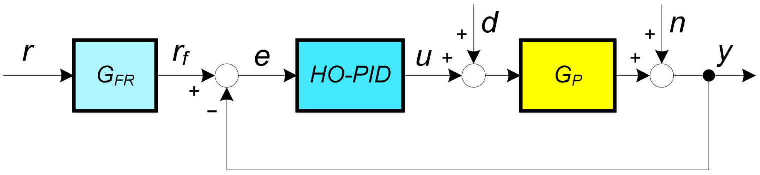


Figure 1. The control loop with HO-PID controller and the process.

Let us now calculate the process closed-loop transfer function $G_{CL}(s)$ from the filtered reference (r_f) to the process output (y). The closed-loop transfer function is then defined as:

$$G_{CL}(s) = \frac{Y(s)}{R_F(s)} = \frac{G_C G_P}{1 + G_C G_P}, \tag{10}$$

where $Y(s)$ and $R(s)$ are the Laplace transforms of the process output and the reference signals, respectively.

When applying the process (6) and the controller (1) transfer functions to (10), and considering that the controller binomial filter is a part of the process (in the process transfer function (6) the areas A_i are replaced by A_{iF}), the closed-loop transfer function becomes:

$$\begin{gathered} G_{CL}(s) = \frac{G_{OL}(s)}{1+G_{OL}(s)}, \\ G_{OL}(s) = A_{0F}K_{-1}s^{-1} + (A_{0F}K_0 - A_{1F}K_{-1}) + s(A_{0F}K_1 - A_{1F}K_0 + A_{2F}K_{-1}) \\ +s^2(A_{0F}K_2 - A_{1F}K_1 + A_{2F}K_0 - A_{3F}K_{-1}) + \cdots \\ +s^k\sum_{i=0}^{k+1}(-1)^i A_{iF}K_{k-i} + \cdots \end{gathered} \tag{11}$$

where G_{OL}(s) denotes the open-loop transfer function $G_{OL}(s) = G_C(s)G_P(s)$.

The MO criteria

According to [35], the MO tuning criterion states that the closed-loop amplitude (magnitude) should be 1 in as wide a frequency bandwidth as possible (starting from frequency $\omega = 0$). This can be achieved if the open-loop transfer function $G_{OL}(j\omega)$, in the Nyquist diagram, follows the vertical line with the real value −0.5 (according to M and N circles in control theory).

Replacing s with complex frequency $j\omega$ in $G_{OL}(s)$ (11) yields:

$$\begin{gathered} G_{OL}(s) = -jA_{0F}K_{-1}\omega^{-1} + (A_{0F}K_0 - A_{1F}K_{-1}) + j\omega(A_{0F}K_1 - A_{1F}K_0 + A_{2F}K_{-1}) \\ -\omega^2(A_{0F}K_2 - A_{1F}K_1 + A_{2F}K_0 - A_{3F}K_{-1}) + \cdots \\ +(j\omega)^k\sum_{i=0}^{k+1}(-1)^i A_{iF}K_{k-i} + \cdots \end{gathered} \tag{12}$$

where j denotes the imaginary component $j = \sqrt{-1}$.

Since merely the real part of the open-loop transfer function is required, only the even powers over frequency in (12) are needed. Therefore:

$$\begin{gathered} \boldsymbol{Re}\{G_{OL}(s)\} = (A_{0F}K_0 - A_{1F}K_{-1}) - \omega^2(A_{0F}K_2 - A_{1F}K_1 + A_{2F}K_0 - A_{3F}K_{-1}) + \cdots \\ +(-1)^q\omega^{2q}\sum_{i=0}^{2q+1}(-1)^i A_{iF}K_{2q-i} + \cdots \end{gathered} \tag{13}$$

In order to achieve that the $\boldsymbol{Re}\{G_{OL}(s)\} = -0.5$ for as high frequencies as possible, the following conditions should be fulfilled:

$$\begin{gathered} -A_{1F}K_{-1} + A_{0F}K_0 = -0.5 \\ -A_{3F}K_{-1} + A_{2F}K_0 - A_{1F}K_1 + A_{0F}K_2 = 0 \\ -A_{5F}K_{-1} + A_{4F}K_0 - A_{3F}K_1 + A_{2F}K_2 - A_{1F}K_3 + A_{0F}K_4 = 0 \\ \vdots \end{gathered} \tag{14}$$

or in matrix form:

$$\begin{gathered} MK_V = C, \ where \\ M_F = \begin{bmatrix} -A_{1F} & A_{0F} & 0 & 0 & 0 & \cdots \\ -A_{3F} & A_{2F} & -A_{1F} & A_{0F} & 0 & \cdots \\ -A_{5F} & A_{4F} & -A_{3F} & A_{2F} & -A_{1F} & \cdots \\ -A_{7F} & A_{6F} & -A_{5F} & A_{4F} & -A_{3F} & \cdots \\ -A_{9F} & A_{8F} & -A_{7F} & A_{6F} & -A_{5F} & \cdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \end{bmatrix}, \ K_V = \begin{bmatrix} A_{-1} \\ K_0 \\ K_1 \\ K_2 \\ K_3 \\ \vdots \end{bmatrix}, \ C = \begin{bmatrix} -0.5 \\ 0 \\ 0 \\ 0 \\ 0 \\ \vdots \end{bmatrix} \end{gathered} \tag{15}$$

Note that the matrix and vector dimensions depend on the number of controller parameters (m + 2):

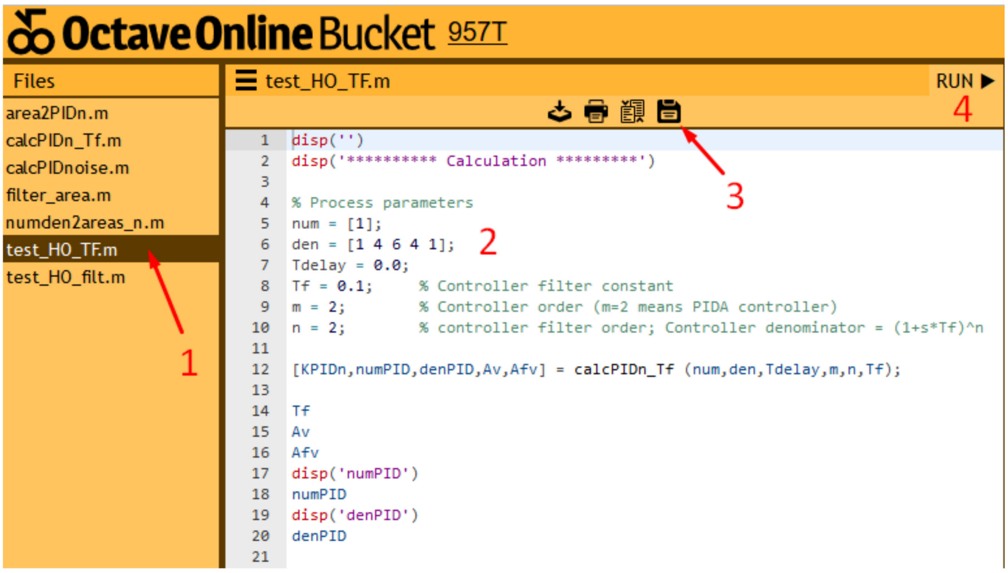
$$M_{(m+2)\times(m+2)}, K_{V_{(m+2)\times 1}}, C_{(m+2)\times 1} \tag{16}$$

The controller parameters (gains) can then be simply calculated from (15):

$$K_V = M^{-1}C \tag{17}$$

The calculation of the controller and filter parameters is straightforward. However, to make it even simpler, we have provided online MATLAB/Octave scripts via the OctaveOnline Bucket website [39]. The provided scripts calculate all the controller parameters for the given process transfer function and the filter time constant. The website layout is shown in Figure 2. The calculation procedure proceeds as follows:

1. Select the appropriate Octave (MATLAB) script (test_HO_TF.m).
2. Provide the process parameters, the filter time constant, the controller order and filter order,
3. press the "Save" button, and
4. press the "Run" button.



```
disp('')
disp('********** Calculation *********')

% Process parameters
num = [1];
den = [1 4 6 4 1];
Tdelay = 0.0;
Tf = 0.1;      % Controller filter constant
m = 2;         % Controller order (m=2 means PIDA controller)
n = 2;         % controller filter order; Controller denominator = (1+s*Tf)^n

[KPIDn,numPID,denPID,Av,Afv] = calcPIDn_Tf (num,den,Tdelay,m,n,Tf);

Tf
Av
Afv
disp('numPID')
numPID
disp('denPID')
denPID

```

Figure 2. The layout of the OctaveOnline Bucket website (function test_HO_TF.m).

The script then calculates the characteristic areas, the controller and the filter parameters. The results are then shown on the right panel of the website.

Illustrative example 1

The D_1^1, PID_2^2 and PID_3^3 controller parameters will be calculated for the following processes:

$$\begin{aligned} G_{P1}(s) &= \frac{1}{(1+s)^4} \\ G_{P2}(s) &= \frac{e^{-0.5s}}{(1+s)^2} \end{aligned} \tag{18}$$

The chosen controller filter time constants is T$_F$ = 0.1. The characteristic areas, without (7), and with controller filter (8) are given in Table 1.

Table 1. The calculated areas for the processes (18) without and with the controller filter.

	A_0	A_1	A_2	A_3	A_4	A_5	A_6	A_7	A_8	A_9
Areas G_{P1}	1	4	10	20	35	56	84	120	165	220
Areas G_{P1} with controller PID_1^1 filter	1	4.1	10.41	21.041	37.104	59.71				
Areas G_{P1} with controller PID_2^2 filter	1	4.2	10.83	22.124	39.317	63.64	96.34	138.63		
Areas G_{P1} with controller PID_3^3 filter	1	4.3	11.26	23.25	41.642	67.81	103.12	148.94	206.66	277.63
Areas G_{P2}	1	2.5	4.125	5.771	7.419	9.068	10.717	12.365	14.014	15.663
Areas G_{P2} with controller PID_1^1 filter	1	2.6	4.385	6.209	8.040	9.872				
Areas G_{P2} with controlle PID_2^2 filter	1	2.7	4.655	6.675	8.708	10.743	12.778	14.814		
Areas G_{P2} with controller PID_3^3 filter	1	2.8	4.935	7.168	9.425	11.685	13.947	16.208	18.470	20.732

The calculated controller parameters (17) are given in Table 2.

Table 2. The calculated controller parameters.

	K_{-1}	K_0	K_1	K_2	K_3
G_{P1}—controller PID_1^1	0.438	1.295	1.041	-	-
G_{P1}—controller PID_2^2	0.812	2.911	3.599	1.556	-
G_{P1}—controller PID_3^3	1.810	7.282	11.008	7.417	1.883
G_{P2}—controller PID_1^1	0.890	1.814	0.934	-	-
G_{P2}—controller PID_2^2	1.140	2.578	1.738	0.300	-
G_{P2}—controller PID_3^3	1.304	3.152	2.459	0.678	0.0674

In order to reduce the excessive swing of the controller output when changing the reference, the following reference filter time constant (9) is used for both processes:

$$T_{FR} = 0.5 \tag{19}$$

Note that the second order reference filter is used (9).

The closed-loop responses for the processes $G_{P1}(s)$ and $G_{P2}(s)$, for all three types of controllers, are shown in Figures 3 and 4. It is clear that tracking and control performance increase by the controller order. Note that the controller output response of PID_3^3 controller is not shown entirely in order to see the responses of PID_1^1 and PID_2^2 controllers more clearly.

When comparing process output responses when using controllers PID_1^1 and PID_3^3 in Figures 3 and 4, it can be seen that the relative difference in performance is larger on higher-order process $G_{P1}(s)$. This is expected, since lower-order processes can already be optimally controlled by lower-order controllers (e.g., the first-order process with PID_0^0 and the second-order process with the PID_1^1 controller). Since the second-order process $G_{P2}(s)$ has an additional delay, the closed-loop performance can still be slightly increased with the PID_3^3 controller.

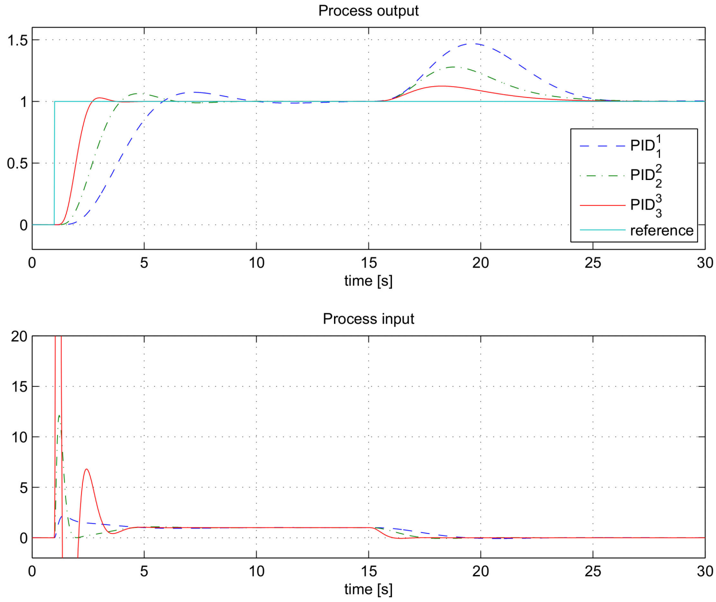


Figure 3. The closed-loop responses for the process G_{P1}(s) when using controllers PID^1_1, PID^2_2 and PID^3_3.

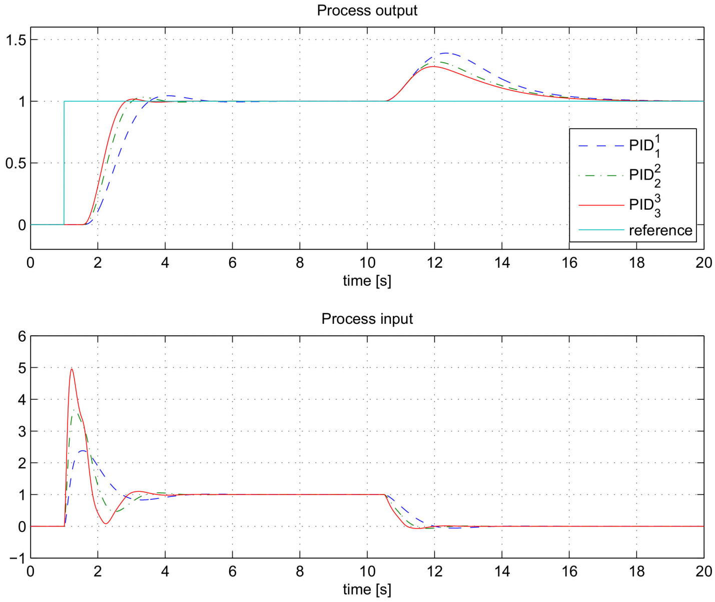


Figure 4. The closed-loop responses for the process $G_{P2}(s)$ when using controllers PID^1_1, PID^2_2 and PID^3_3.

According to the closed-loop responses, it can be concluded that HO-PID controllers can significantly improve the closed-loop performance, especially for higher-order processes. The only required parameter from the user is the controller filter time constant T_F. Namely, the amplification of the process output measurement noise depends on the chosen T_F. However, the relation between T_F and the actual amplification of the high-frequency (HF) noise depends on several other controller parameters and is a rather complex function. Therefore, in practice, it would be more appropriate to define the desired HF noise amplification than the controller filter time constant.

3. HO-PID Controller with Specified HF Noise Amplification

As mentioned in the previous section, the n-th order controller filter $G_F(s)$ (1) is primarily used to decrease the controller output noise due to the measurement noise (in addition to making the entire controller transfer function proper or strictly proper and, therefore, realisable in practice). High controller amplification of the measurement noise is never desired, since it may also cause large swings of the control output signals and thus may decrease the actuator's life span. In order to limit the amplification of the process measurement noise, the user can try different values of T_F until the desired amplification (attenuation) of the noise is achieved. In practice, this may take too long, since the function between T_F and the noise amplification is complex and non-linear. Therefore, from the user's perspective, it is easier to define the desired noise amplification of the controller than select filter time constant T_F.

The process output noise (n_y) is amplified by the controller (1) in the closed-loop configuration as follows:

$$U_N = G_{CN}(s)N_y = \frac{G_{CF}(s)}{1+G_{CF}(s)G_P(s)}N_y = \left(G_P(s) + \frac{1}{G_{CF}(s)}\right)^{-1} N_y, \tag{20}$$

where N_y and U_N are Laplace transforms of the measurement noise and the controller output noise, respectively. The negative sign is omitted to simplify the derivation. From (20) it can be seen that at lower frequencies, the transfer function $G_{CN}(s)$ is mostly dominated by the process transfer function $G_P(s)$, while at higher frequencies, it is mostly dominated by the controller transfer function $G_{CF}(s)$. At lower frequencies, the process can be approximated by its gain K_{PR}, while at higher frequencies the controller gains K_{-1} and K_0 can be neglected. Therefore, $G_{CN}(s)$ can be approximated by the following transfer function:

$$G_{CN}(s) \approx \frac{K_{PR}^{-1} + K_1 s + K_2 s^2 + \cdots + K_m s^m}{(1+T_F s)^n}. \tag{21}$$

On the other hand, the desired controller output noise (U_{ND}) should be similar to:

$$U_{ND} = K_{HF} N_y, \tag{22}$$

where K_{HF} is a chosen noise amplification factor. Since amplitudes U_N and U_{ND} cannot be compared directly due to different frequency characteristics, it is easier to compare noise powers of both signals in some chosen frequency bandwidth (from ω_1 to ω_2). Namely, due to Parseval theorem, the power of the controller output signal (P_{UN}) is proportional to:

$$P_{UN} \propto \int_{\omega_1}^{\omega_2} \left|G_{CN}(\omega)N_y(\omega)\right|^2 d\omega\ . \tag{23}$$

The desired noise power is, according to (22), proportional to:

$$P_{UND} \propto \int_{\omega_1}^{\omega_2} \left|K_{HF}N_y(\omega)\right|^2 d\omega\ . \tag{24}$$

When considering the N_y as a white noise with amplitude over frequency as $N_y(\omega)$ = 1, the powers P_{UN} and P_{UND} would become the same when:

$$\begin{gathered}
\int_{\omega_1}^{\omega_2} \frac{F_0+F_1\omega^2+F_2\omega^4+\cdots+F_m\omega^{2m}}{\left(1+T_F^2\omega^2\right)^n} d\omega = K_{HF}^2(\omega_2-\omega_1), where\ K_0 = K_{PR}^{-1}\ and \\
F_0 = K_0^2 \\
F_1 = K_1^2 - 2K_0K_2 \\
\vdots \\
F_k = K_k^2 + 2\sum_{i=0}^{k-1}(-1)^{k+i}K_iK_{2k-i} \\
\vdots \\
F_m = K_m^2
\end{gathered} \tag{25}$$

However, the solution of the integral in (25), due to the denominator (controller filter), becomes very complex and highly non-linear in respect to T_F. Therefore, some search algorithm (optimization) must be applied for each calculation of the T_F.

This would seriously impact the simplicity of the proposed method. Therefore, it is decided to simplify the function inside the above integral. Since at higher frequencies, the most dominant controller term becomes the one with the highest derivative (K_m), the function can be simplified as follows:

$$\frac{F_0 + F_1\omega^2 + F_2\omega^4 + \cdots + F_m\omega^{2m}}{\left(1 + T_F^2\omega^2\right)^n} \approx \begin{cases} F_m\omega^{2m} & ;\omega \le \frac{1}{T_F} \\ \frac{F_m\omega^{2(m-n)}}{T_F^{2n}} & ;\omega > \frac{1}{T_F} \end{cases}. \tag{26}$$

According to (26), the expression (25) simplifies into:

$$\begin{gathered} \int_{\omega_1}^{\omega_F} F_m\omega^{2m} d\omega + \int_{\omega_F}^{\omega_2} F_m\omega^{2(m-n)}\omega_F^{2n} d\omega \approx K_{HF}^2(\omega_2 - \omega_1) \\ \int_{\omega_1}^{\omega_F} F_m\omega^{2m} d\omega = \frac{F_m\left(\omega_F^{2m+1} - \omega_1^{2m+1}\right)}{(2m+1)} \\ \int_{\omega_F}^{\omega_2} F_m\omega^{2(m-n)}\omega_F^{2n} d\omega = \begin{cases} \frac{F_m\omega_F^{2n}\left(\omega_2^{2(m-n)+1} - \omega_F^{2(m-n)+1}\right)}{(2m-2n+1)} & n > m \\ F_m\omega_F^{2n}(\omega_2 - \omega_F) & n = m \end{cases} \\ \omega_F = \frac{1}{T_F} \end{gathered} \tag{27}$$

The contribution of noise power in the frequency region below ω_F is usually much smaller than at higher frequencies. Therefore, in order to even further simplify the derivation, we can choose $\omega_1 = 0$ without making any significant error in the calculation. Selection of the upper frequency (ω_2) in the integral, due to the Shannon theorem, depends on the controller sampling frequency. Without loss of generality, the upper frequency can be selected as:

$$\omega_2 = \omega_S = \frac{2\pi}{T_S}\ , \tag{28}$$

where ω_S is controller sampling frequency (in rad/s) and T_S is controller sampling time. By taking into account that:

$$\omega_S \gg \omega_F, \tag{29}$$

and $\omega_1 = 0$, the expression (27) simplifies even further:

$$\begin{gathered} \int_0^{\omega_F} F_m\omega^{2m} d\omega = \frac{F_m\omega_F^{2m+1}}{2m+1} \\ \int_{\omega_F}^{\omega_S} \frac{F_m\omega^{2(m-n)}}{T_F^{2n}} d\omega \approx \begin{cases} \frac{F_m\omega_F^{2m+1}}{2(n-m)-1} & n > m \\ F_m\omega_F^{2n}\omega_S & n = m \end{cases} \end{gathered} \tag{30}$$

Therefore, the final expression, when taking into account that $F_m = {K_m}^2$, reads as:

$$\begin{array}{cc} K_m^2\omega_F^{2m+1}\left(\frac{1}{2m+1} + \frac{1}{2(n-m)-1}\right) \approx K_{HF}^2\omega_S & n > m \\ K_m^2\omega_F^{2m} \approx K_{HF}^2 & n = m \end{array}, \tag{31}$$

Therefore, the filter time constant ($T_F = 1/\omega_F$) can be estimated as follows:

$$\begin{array}{ll} T_F \approx \sqrt[2m+1]{\frac{K_m^2\left(\frac{1}{2m+1} + \frac{1}{2(n-m)-1}\right)}{K_{HF}^2\omega_S}} & ;\ n > m \\ T_F \approx \sqrt[m]{\frac{|K_m|}{K_{HF}}} & ;\ n = m \end{array}, \tag{32}$$

Note that the above derivation of the filter time constant takes into account approximations (26) and (29). This means that the final output noise power of the controller may differ from that defined by the selected high frequency gain K_{HF}. However, if the above approximations are not taken into account, then the final expressions for the calculation of ω_F in (31) would become those of the higher order without analytic solution for T_F. This would significantly complicate the filter calculation.

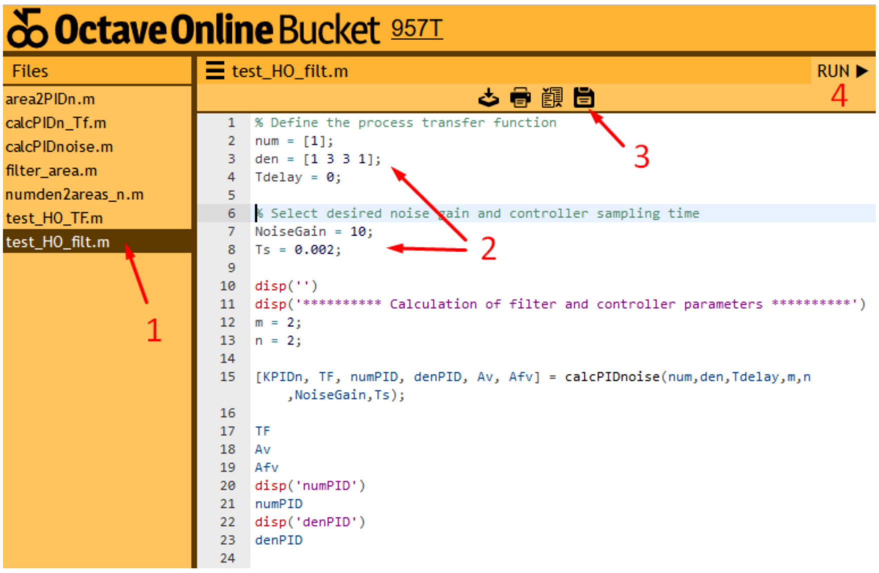
The entire procedure for the calculation of the controller parameters for a given process is given in Figure 5.

1. Calculate the moments from the given G$_P$(s) (7) or from the process response in time domain (Vrančić et al., 2001; Vrančić, 2011).
2. Choose some initial filter time constant T_F.
3. For a given filter order (n), calculate modified areas according to the chosen T_F from (8).
4. For a given model order (m) calculate the controller parameters from (17).
5. Calculate constant T_F from (32).
6. Repeat steps 3–5 a few times (3 times is usually enough) until value T_F settles.

Figure 5. Calculation of the filter and controller parameters.

As shown in Figure 5, the calculation of the controller and filter parameters is straightforward. However, to make it even simpler, as mentioned before, we have provided online MATLAB/Octave scripts via OctaveOnline Bucket website [39]. The website layout is shown in Figure 6. The calculation procedure proceeds as follows:

1. Select the appropriate Octave (MATLAB) script (test_HO_filt.m).
2. Provide the process parameters, the desired noise gain, the controller sampling time, the controller order and filter order,
3. press the "Save" button, and
4. press the "Run" button.



```
% Define the process transfer function
num = [1];
den = [1 3 3 1];
Tdelay = 0;

% Select desired noise gain and controller sampling time
NoiseGain = 10;
Ts = 0.002;

disp('')
disp('********** Calculation of filter and controller parameters **********')
m = 2;
n = 2;

[KPIDn, TF, numPID, denPID, Av, Afv] = calcPIDnoise(num,den,Tdelay,m,n
    ,NoiseGain,Ts);

TF
Av
Afv
disp('numPID')
numPID
disp('denPID')
denPID

```

Figure 6. The layout of the OctaveOnline Bucket website (function test_HO_filt.m).

The script then calculates the filter time constant, the characteristic areas and the controller parameters. The results are shown on the right panel of the website.

Illustrative example 2

Consider the following fourth-order process transfer function $G_{P3}(s)$ (18):

$$G_{P3}(s) = \frac{e^{-0.2s}}{(1+s)^4} \tag{33}$$

The initially chosen controller filter time constants is T_F = 0.1. For all the experiments in this section, the chosen sampling time is T_S = 0.002 s. In order to retain clarity of the derivations, the characteristic areas of the process are not mentioned herein; however, they can be calculated (besides all the controller and the filter parameters) on the aforementioned website [39].

a. Changing the parameter K_{HF}

According to the procedure given in Figure 5, when choosing parameter K_{HF}, controller PID_4^3 and repeating steps 3–5 a few times (in our case, 3 times), the calculated filter time constants, and the calculated controller parameters (17) are given in Table 3.

Table 3. The calculated filter time constants T_F and controller parameters at different noise gains K_{HF}.

	T_F	K_{-1}	K_0	K_1	K_2	K_3	σ_{rel}
K_{HF} = 2	0.209	0.679	2.445	3.282	1.954	0.440	1.85
K_{HF} = 5	0.155	0.773	2.684	3.426	1.896	0.382	4.60
K_{HF} = 10	0.124	0.844	2.873	3.562	1.885	0.352	9.2
K_{HF} = 20	0.100	0.914	3.062	3.710	1.890	0.329	18.3

Again, in order to reduce the excessive swing of the controller output when changing the reference, the following second-order reference filter time constant (9) is used:

$$T_{FR} = 0.2 \tag{34}$$

The closed-loop responses for different values of K_{HF} are given in Figure 7.

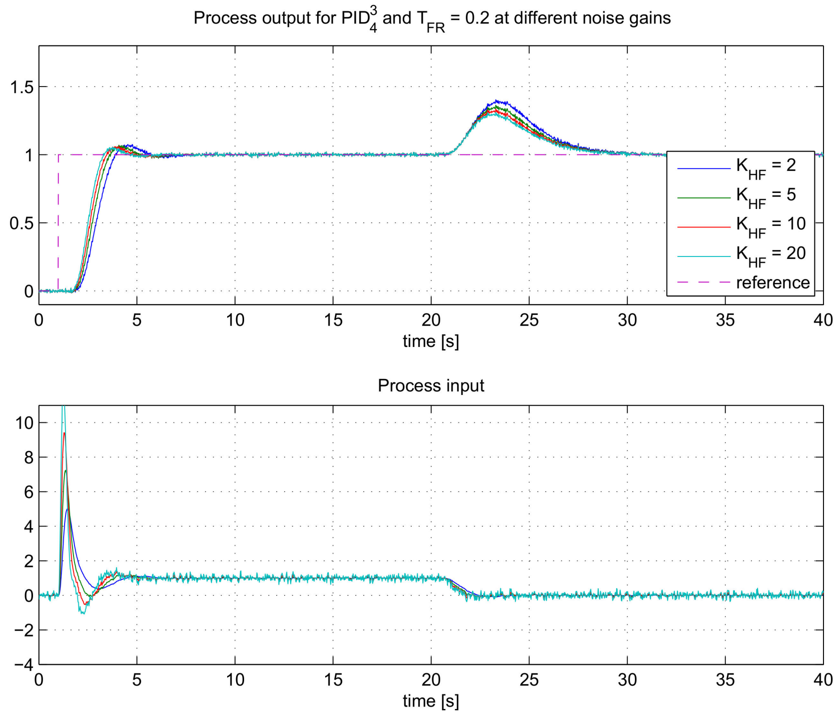


Figure 7. The closed-loop responses for the process $G_{P3}(s)$ when using controllers PID_4^3, at different K_{HF}.

As expected, the speed of the closed-loop response and the controller output signal noise increases by increasing the noise gain factor K_{HF}. However, the improvement of the closed-loop speed is not so significant at the highest factors K_{HF}. On the other hand, the controller output noise increases at higher factors K_{HF}. As expected, there is a trade-off between the closed-loop speed and the amount of the controller output noise. Therefore, in practice, the allowed noise gain should be chosen wisely according to the amount of noise present in the system.

The actual "amplification" of the measurement noise (the actually achieved noise gain K_{HF}) is measured by dividing standard deviations of the controller output signal (u) and the process output (n) when the process is in the steady-state:

$$\sigma_{rel} = \frac{\sigma_u}{\sigma_y}. \tag{35}$$

The actual amplifications of the measurement noise signals are given in Table 3. It is obvious that the actual gains of the noise (σ_{rel}) are very similar to the desired ones (K_{HF}).

b. Changing the filter order (n)

On the other hand, the speed of the closed-loop response, for the same K_{HF} and the controller order (m), can also be altered by changing the filter order. In this regard, we tested the performance of the controllers PID_n^3, where n varies from 3 to 6.

According to the procedure given in Figure 5, when choosing K_{HF} = 10 and repeating steps 3–5 3 times, the calculated filter time constants and the controller parameters (17) are given in Table 4.

Table 4. The calculated filter time constants T_F and controller parameters at different n.

Controller Structure	T_F	K_{-1}	K_0	K_1	K_2	K_3	σ_{rel}
PID_3^3	0.396	0.619	2.402	3.547	2.388	0.629	10.1
PID_4^3	0.124	0.844	2.873	3.562	1.885	0.352	9.17
PID_5^3	0.109	0.799	2.731	3.410	1.824	0.348	5.05
PID_6^3	0.104	0.742	2.564	3.247	1.777	0.352	3.33

The closed-loop responses for different controller filter orders (n = 3 to 6) are given in Figure 8.

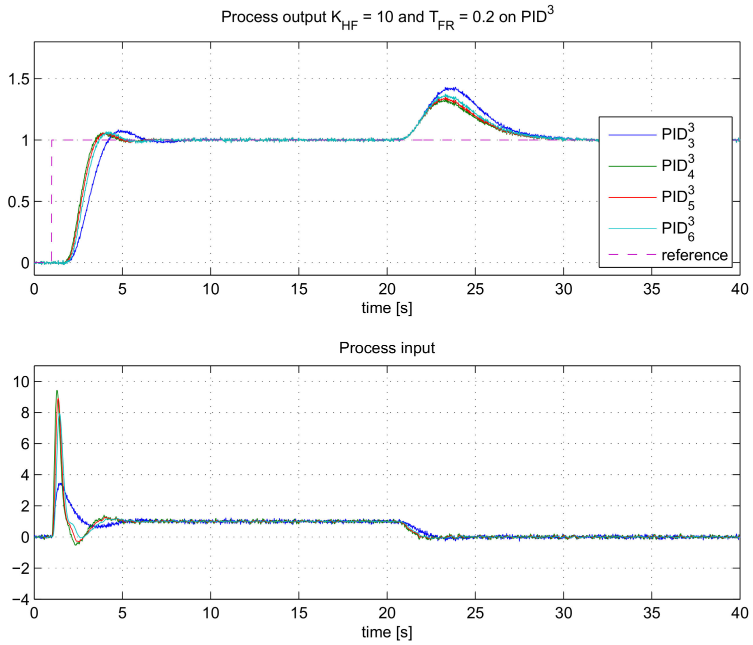


Figure 8. The closed-loop responses for the process $G_{P3}(s)$ when using controllers with different filter orders PID_n^3, at K_{HF} = 10.

As can be seen, the speed of the closed-loop response is the highest for controller PID_4^3. The speed of controllers with higher-order filters (n > 4) are slightly slower. The speed of response for n > 4 is not improving since a higher-order filter also adds some complexity to the closed-loop transfer function. This, in return, may result in lower closed-loop speeds.

The practical question is how to find the most optimal controller filter order in advance, before making the closed-loop experiment on the process. This can be answered by calculating the integral of control error (IE), which can be considered as a measure of the closed-loop speed:

$$IE = \int_{t=0}^{\infty} (r - y)dt \tag{36}$$

If the closed-loop responses have small overshoots, the higher values of IE indicate slower closed-loop responses. For such responses, the IE can be a useful tool to measure the closed-loop speed. The IE value can be relatively easily calculated by transforming the Equation (36) into Laplace domain. It can be shown that:

$$IE = \frac{1}{K_{PR}K_{-1}}. \tag{37}$$

Therefore, for the process with the same steady-state gain K_{PR} (2), the closed-loop speed is inversely proportional to the integrating gain (K_{-1}) of the controller. Therefore, the controller with the highest gain K_{-1} will produce the fastest closed-loop response (providing that the closed-loop responses have small or negligible overshoots). Indeed, from Table 4 it is evident that the highest gain K_{-1} is calculated for controller PID_4^3. This corresponds to our previous observations.

The actual amplifications of the measurement noise signals, according to (35), are given in Table 4. The actual noise gains (σ_{rel}) are very similar to the desired ones (K_{HF}) for filter orders 3 and 4, while for higher-order filters the actual noise gain is lower. This is due to various assumptions (simplifications) made when deriving the filter time constant (32).

c. Changing the controller order (m)

As is already known, the speed of the closed-loop response can also be altered by changing the controller order. In this regard, we tested the performance of the controllers PID_n^m, where m varies from 1 to 4. In all cases the controller filter is chosen to be 1 order higher than the controller order (n = m + 1). The desired noise gain remains the same as in the previous experiment (K_{HF} = 10).

The calculated filter time constants and the controller parameters (17) are given in Table 5.

Table 5. The calculated filter time constants T_F and controller parameters at different controller order m (n = m + 1).

Controller Structure	T_F	K_{-1}	K_0	K_1	K_2	K_3	K_3	σ_{rel}
PID_2^1	0.015	0.517	1.323	0.904	-	-	-	10.9
PID_3^2	0.078	0.766	2.36	2.45	0.863	-	-	9.7
PID_4^3	0.124	0.844	2.873	3.562	1.885	0.352	-	9.2
PID_5^4	0.161	0.876	3.270	4.694	3.225	1.068	0.143	8.9

The closed-loop responses for different controller orders (m = 1 to 4) are given in Figure 9.

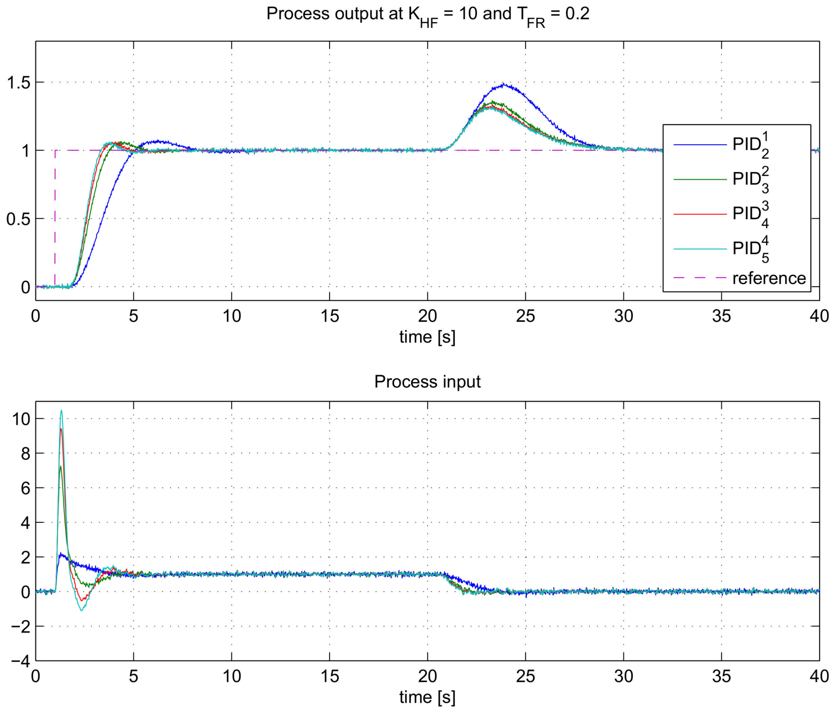


Figure 9. The closed-loop responses for the process $G_{P3}(s)$ when using controllers with different controller orders PID_n^m, at K_{HF} = 10 and n = m + 1.

As can be seen, the closed-loop speed increases by increased controller order, similar to the results in Figures 3 and 4. The difference is that now the controller output noise is under control (K_{HF} = 10), so the level of control noise is similar for all of the controllers. The fastest responses are obtained with PID_5^4. In a similar manner as in the previous case, the speed of responses can be estimated by comparing the values of the calculated integrating gains (K_{-1}) in Table 5. Indeed, K_{-1} is the highest for PID_5^4.

The actual amplifications of the measurement noise signals, according to (35), are given in Table 5. The actual gains of the noise (σ_{rel}) are very similar to the desired ones (K_{HF}) for all controller orders.

4. Robustness

The proposed design of HO-PID controllers results in a relatively fast and non-oscillatory response. In addition, the controller noise is under control by choosing parameter K_{HF}. However, the designed closed-loop system can still be not robust enough to process variations. Namely, due to nonlinearity or time-variations of the process, its characteristics (gain, delay, time constants, etc.) can vary by working point or by time.

The robustness of a stable closed-loop system is usually measured by maximum sensitivity (M_S) [1,38]. Maximum sensitivity is related to the distance of the open-loop transfer function $G_C(j\omega)G_P(j\omega)$ from the critical point (−1+j0). Namely, M_S is the inverse of the minimum distance between the open-loop transfer function and the critical point. Generally, a smaller value of M_S denotes a more robust closed-loop system to process variations. Usual values of M_S for stable processes are between 1.4 and 2.0 [1,38].

The robustness of the closed-loop system for the proposed HO-PID controllers has been tested on the following third-order process with delay:

$$G_{P4}(s) = \frac{K_{PR}e^{-T_{del}s}}{(1+sT)^3}, \tag{38}$$

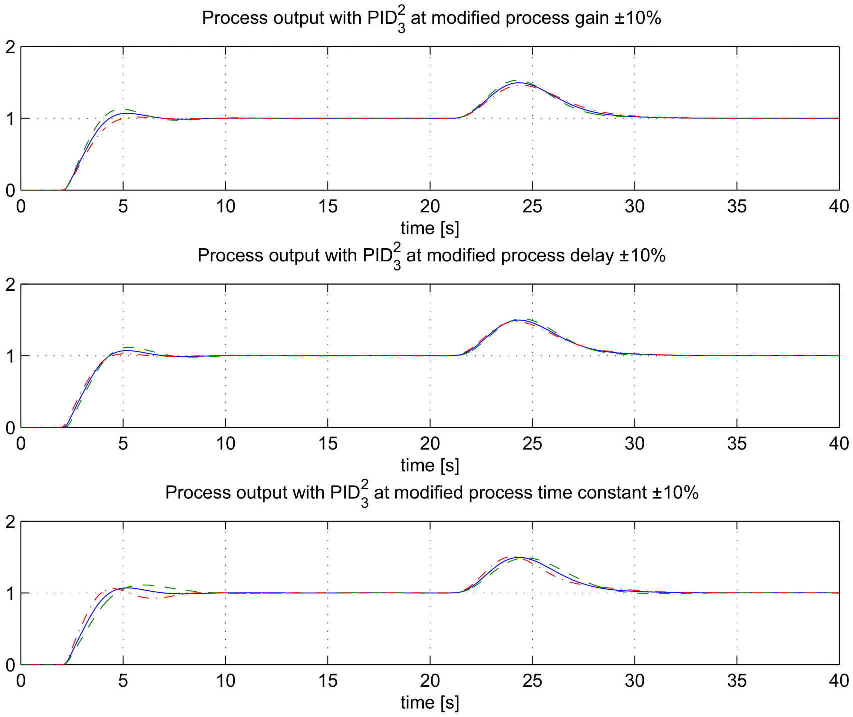
where the nominal values are K_{PR} = 1, T_{del} = 1 and T = 1. Three different HO-PID controllers are selected: PID_3^2, PID_4^3, and PID_5^4. The calculated controller parameters, when choosing

K_{HF} = 10, T_S = 0.002 s and according to the proposed tuning method, are shown in Table 6. The calculated values of the maximum sensitivity M_S, for all three controllers, are shown in the same table. It can be seen that the M_S values slightly increase with the increased controller order. However, the differences are not large and all the values are below 2.0.

Table 6. The calculated filter time constants T_F and controller parameters at different controller orders m ($n = m + 1$) for $G_{P4}(s)$.

Controller Structure	T_F	K_{-1}	K_0	K_1	K_2	K_3	K_3	M_S
PID_3^2	0.0702	0.498	1.597	1.752	0.668	-	-	1.78
PID_4^3	0.125	0.554	1.992	2.674	1.596	0.362	-	1.86
PID_5^4	0.170	0.579	2.308	3.629	2.832	1.117	0.185	1.91

Besides calculating the M_S values, we were also simulating the closed-loop responses using all three controllers on the nominal process, and on the changed process (±10% change of process gain K_{PR}, time-delay T_{del} and time constant T). The closed-loop responses are shown in Figures 10–12. It is evident that the closed-loop responses under perturbed parameters are still stable without significant oscillations. When comparing Figures 10 and 12 it can be noticed that the perturbed parameters with controller PID_5^4 result in slightly more deviation from the nominal response than with controller PID_3^2. This is all in accordance with the calculated values of M_S in Table 6.


Process output with PID$_3^2$ at modified process gain ±10%

Process output with PID$_3^2$ at modified process delay ±10%

Process output with PID$_3^2$ at modified process time constant ±10%

Figure 10. The closed-loop responses for the process $G_{P4}(s)$, using controller PID_3^2 at K_{HF} = 10 for nominal (solid line), 10% increased (dashed line) and 10% decreased (dash-dotted line) process parameters.

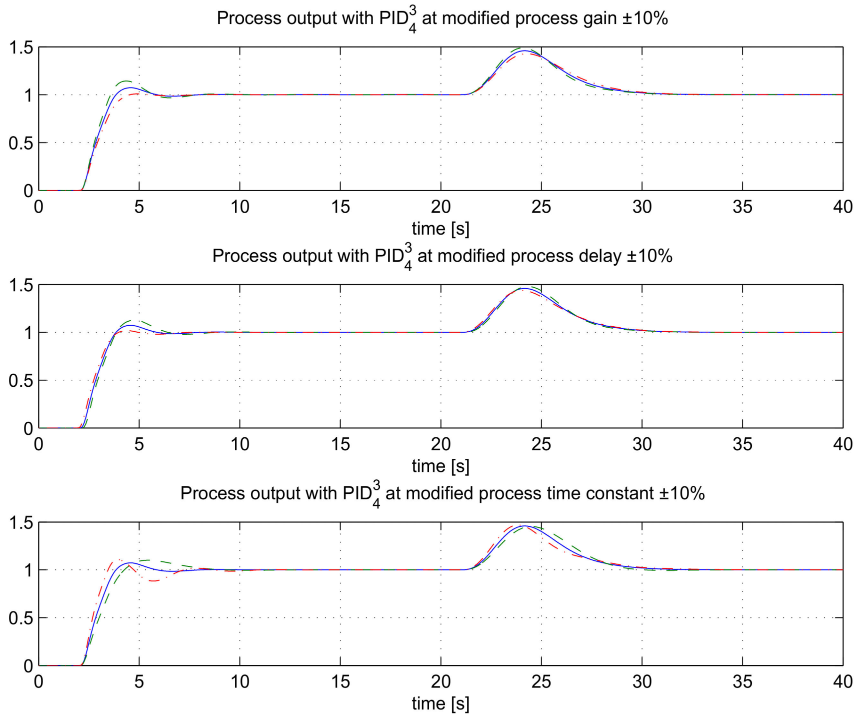


Figure 11. The closed-loop responses for the process $G_{P4}(s)$, using controller PID_4^3 at $K_{HF} = 10$ for nominal (solid line), 10% increased (dashed line) and 10% decreased (dash-dotted line) process parameters.

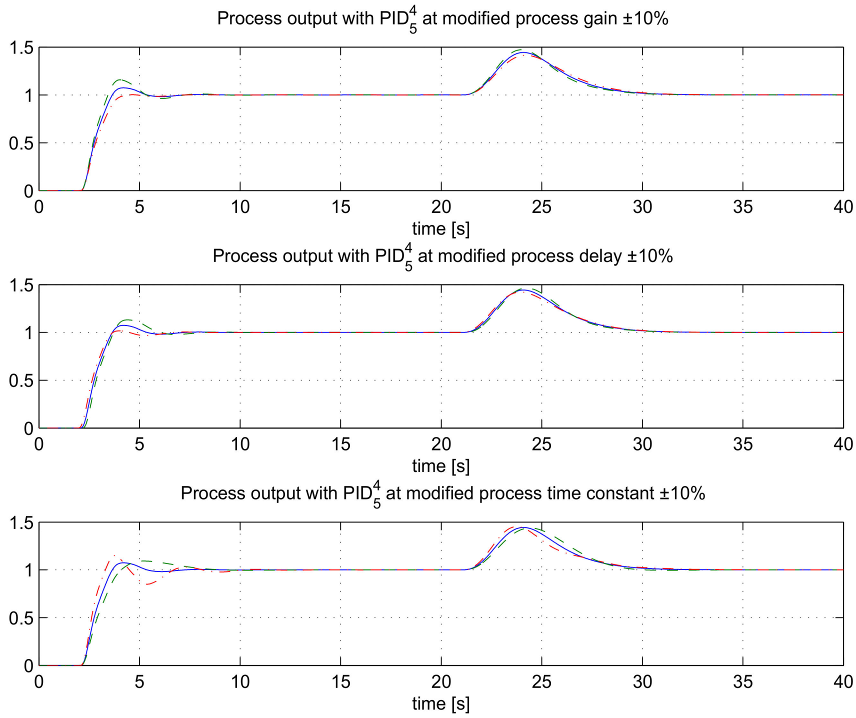


Figure 12. The closed-loop responses for the process $G_{P4}(s)$, using controller PID_5^4 at $K_{HF} = 10$ for nominal (solid line), 10% increased (dashed line) and 10% decreased (dash-dotted line) process parameters.

5. Comparison with Other Tuning Methods

The proposed tuning method was compared with some other methods for PIDA controllers. The chosen methods, which were tested on a particular process model, are from

Lurang and Puangdownreong [21] (denoted as the Lurang method from here on) and Jung and Dorf [11] (denoted as the Jung method from here on). The Lurang method involves calculating the PIDA controller parameters by optimizing the tracking and disturbance rejection response under several limitations given on rise time, overshoot, settling time, steady-state error and similar. The optimization is carried out with a modified bat algorithm proposed by the authors. The Jung method analytically calculates the PIDA controller parameters for the third-order process according to provided desired overshoot and settling time. Both methods do not take into account the controller's filter. Therefore, the actual implementation in practice could be questionable if the filter dynamics become slower.

Case 1

The following process model has been selected, according to [21]:

$$G_{P5}(s) = \frac{1}{(1+s)(1+0.5s)\left(1+\frac{s}{3}\right)} \tag{39}$$

The Lurang method suggests the following PIDA controller parameters:

$$K_{-1} = 2.20,\ K_0 = 3.60,\ K_1 = 1.60,\ K_2 = 0.06 \tag{40}$$

The chosen controller filter time constant was very low (T_F = 0.01), since we did not want to spoil the closed-loop response of the Lurang method. Namely, as already mentioned, the Lurang method does not take into account the controller filter in the design phase.

For comparison, we chose the controller structures with the lowest possible controller filter order n: PID_2^2. For illustrative purposes, the one-order higher controller structure (PID_3^3) was also tested. Note that the closed-loop results of our proposed method, for the same level of controller noise, can be improved by using $n > m$.

The calculated controller parameters, for the given process and controller filter were the following:

$$\begin{aligned} PID_2^2:\ & K_{-1} = 25.06,\ K_0 = 45.95,\ K_1 = 25.07,\ K_2 = 4.18 \\ PID_3^3:\ & K_{-1} = 37.53,\ K_0 = 69.42,\ K_1 = 38.88,\ K_2 = 6.88,\ K_3 = 0.104 \end{aligned} \tag{41}$$

We tested, separately, the tracking response and the disturbance rejection when using all three controllers. For tracking response, the reference (r) changed from 0 to 1 at t = 1 s and for disturbance response the process input disturbance (d) changed from 0 to 1 at t = 1 s.

The closed-loop responses are shown in Figure 13.

As can be seen, the responses of the proposed method with PID_2^2 controller are superior to the Lurang method. Certainly, the one-order higher controller (PID_3^3) has a better result.

For a more objective comparison, the integral of squared error (ISE) signal has been calculated for all three controllers. The results are shown in Table 7. It is obvious that the ISE values for the PID_2^2 controller are much lower than the ones for the Lurang controller.

Table 7. The ISE values for all three controllers in tracking and disturbance rejection.

ISE	PID_2^2	PID_3^3	Lurang
Tracking	0.0228	0.0131	0.322
disturbance rejection	4.38×10^{-7}	1.95×10^{-7}	0.061

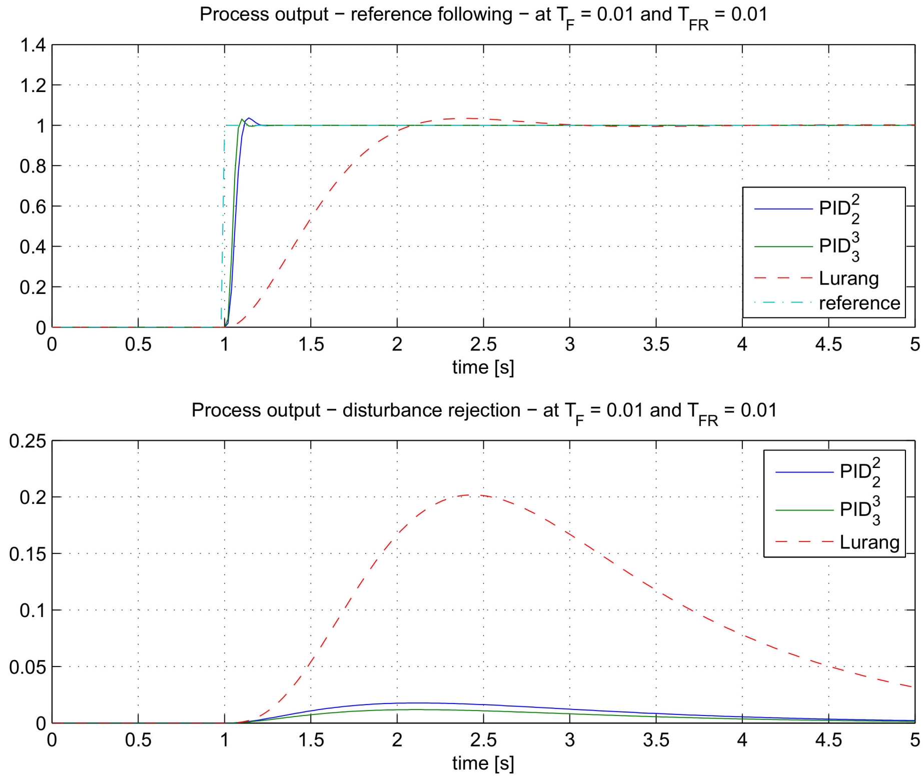


Figure 13. The comparison of the closed-loop responses for the process $G_{P5}(s)$ when using PID_2^2, PID_3^3 and the Lurang controller.

Case 2

The second process model has been selected according to [11]:

$$G_{P6}(s) = \frac{0.0556}{(1+s)\left(1+\frac{s}{3}\right)\left(1+\frac{s}{6}\right)} \tag{42}$$

The Jung method suggests the following PIDA controller parameters:

$$K_{-1} = 529.8,\ K_0 = 516.5,\ K_1 = 179.2,\ K_2 = 26.3 \tag{43}$$

As before, the controller filter time constant was chosen very low (T_F = 0.005), since we wanted to preserve the closed-loop response of the Jung method, which was obtained without the controller filter.

The calculated PID_2^2 and PID_3^3 controller parameters, for the given process and controller filter were the following:

$$\begin{aligned} PID_2^2:\ K_{-1} = 902,\ K_0 = 1353,\ K_1 = 501,\ K_2 = 50.1 \\ PID_3^3:\ K_{-1} = 1351,\ K_0 = 2037,\ K_1 = 767,\ K_2 = 81.3,\ K_3 = 0.6 \end{aligned} \tag{44}$$

As in the previous case, the closed-loop responses were tested on the tracking response and the disturbance rejection. The closed-loop responses are shown in Figure 14.

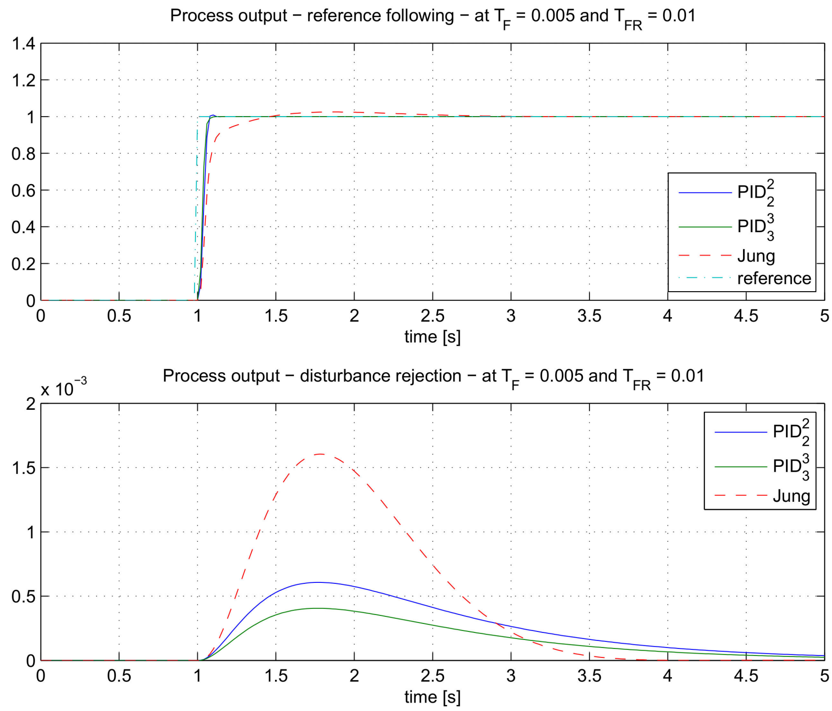


Figure 14. The comparison of the closed-loop responses for the process $G_{P6}(s)$ when using PID_2^2, PID_3^3 and the Jung controller.

Again, the responses of the proposed method with PID_2^2 and PID_3^3 controllers are superior to the Jung method. The comparison of ISE values in Table 8 shows PID_2^2 controller has lower values than the Jung controller. However, note that the disturbance rejection settling time is the best with the Jung method.

Table 8. The ISE values for all three controllers in tracking and disturbance rejection.

ISE	PID_2^2	PID_3^3	Jung
Tracking	8.12×10^{-3}	4.11×10^{-3}	1.97×10^{-2}
disturbance rejection	4.38×10^{-7}	1.95×10^{-7}	2.18×10^{-6}

Case 3

The fourth-order process model has been selected according to [6]:

$$G_{P7}(s) = \frac{1}{(1+s)\left(1+\frac{s}{2}\right)\left(1+\frac{s}{4}\right)\left(1+\frac{s}{8}\right)} \tag{45}$$

The Puangdownreong method suggests the following PIDA controller parameters:

$$K_{-1} = 1.647,\ K_0 = 2.684,\ K_1 = 1.105,\ K_2 = -2.65 \cdot 10^{-3} \tag{46}$$

The method calculated the following controller filter:

$$G_F(s) = \frac{1}{1 + 0.0132s + 5.26 \cdot 10^{-5} s^2} \tag{47}$$

which was also used in design of the proposed PID_2^2 controller parameters. For the given process and controller filter transfer function, the following controller parameters were calculated:

$$PID_2^2:\ K_{-1} = 4.36,\ K_0 = 7.735,\ K_1 = 3.97,\ K_2 = 0.60 \tag{48}$$

As in the previous case, the closed-loop responses were tested on the tracking response and the disturbance rejection. The closed-loop responses are shown in Figure 15.

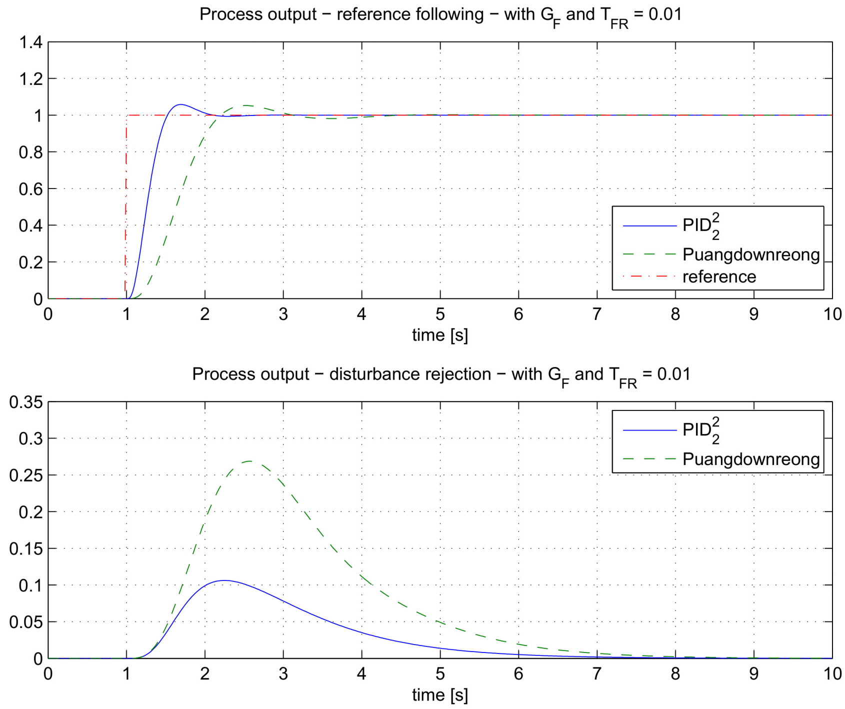


Figure 15. The comparison of the closed-loop responses for the process G_{P6}(s) when using PID_2^2 and the Puangdownreong controller.

Again, the responses of the proposed method with the PID_2^2 controller are superior to the Puangdownreong method. The comparison of ISE values in Table 9 shows the PID_2^2 controller has lower values than the Puangdownreong controller.

Table 9. The ISE values for both controllers in tracking and disturbance rejection.

ISE	PID_2^2	Puangdownreong
Tracking	0.172	0.466
disturbance rejection	0.0162	0.107

6. Conclusions

In the paper, the method for tuning the parameters of the m-th order controller with the n-th order binomial filter has been presented. The proposed tuning method is based on the MO criteria which aims to produce non-oscillatory and fast closed-loop reference step responses. The calculation of the controller parameters is analytical and does not require any kind of optimization. An additional advantage of the proposed method is that the process can be described either by the process model or by the process time responses during the steady-state change.

To keep the noise gain of the controller under control, the filter time constant of the controller can also be calculated according to the specified noise gain. The calculation procedure is still analytical, and the results confirm that the level of controller noise is consistent with the given noise gain. The only exception is the use of larger relative degrees between the controller and the filter order, which is the consequence of some simplifications in the calculation of the filter time constant.

The proposed method was tested on six different process models (from second to fourth-order process models with or without time delay). The results confirmed that the control performance can be improved by increasing the controller order or by selecting the filter order appropriately without increasing the controller output noise. The study shows that increasing the filter order improves the performance only up to a certain level, after which the performance starts to decrease. The optimum degree of controller and filter order can be easily determined by the value of integral gain of the controller.

The tuning method was compared with three other tuning methods for PIDA controllers (Lurang [21], Jung [11] and Puangdownreong [6] methods). Although the selected process models were the same as in the aforementioned methods, the proposed method resulted in a better control performance.

Therefore, the proposed higher-order controller design is efficient, and the controller output noise gain is under control. However, it does not mean that the proposed method cannot be improved. Indeed, the proposed method is based on optimizing the reference tracking performance. In our further research, we plan to improve the disturbance rejection performance as well. Namely, the article shows that the MO controller design leads to a strong asymmetry in the dynamics of tracking and disturbance rejection behaviour. While the HO-PID controller design leads to an increase in the number of pulses of the control signal after reference step changes, the responses of the control signal after the change of disturbance remain monotonic. This motivates us to deal with the modification of the MO controller design with regard to a faster response to disturbances.

Moreover, we also plan to design a method that will find the most optimal controller and filter order for the given process and noise amplification considering the complexity of the controller and filter order. We plan to calculate the optimal parameters of the reference filter to control the change of the controller output signal when the reference signal is changed.

Another planned modification is adding a user-defined parameter for changing the speed of the closed-loop control. Slowing down the control speed would further increase the robustness of the system.

Author Contributions: Writing-original draft preparation, D.V. and M.H. Simulations, D.V. and M.H. Editing, D.V. and M.H. Project administration, D.V. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the grants P2-0001 financed by the Slovenian Research Agency, APVV SK-IL-RD-18-0008 platoon modelling and control for mixed autonomous and conventional vehicles: a laboratory experimental analysis, and VEGA 1/0745/19 control and modelling of mechatronic systems in e-mobility.

Acknowledgments: Supported by Slovenská e-akadémia, n. o.

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

References

1. Åström, K.J.; Hägglund, T. *PID Controllers: Theory, Design, and Tuning*, 2nd ed.; Instrument Society of America: Pittsburgh, PA, USA, 1995.
2. Vilanova, R.; Visioli, A. *PID Control in the Third Millennium*, 1st ed.; Vilanova, R., Visioli, A., Eds.; Advances in Industrial Control; Springer: London, UK, 2012; ISBN 978-1-4471-2424-5.
3. Alfaro, V.M.; Vilanova, R. *Model-Reference Robust Tuning of PID Controllers*; Springer International Publishing: Cham, Switzerland, 2016. [CrossRef]
4. Visioli, A. *Practical PID Control*; Springer: London, UK, 2016. [CrossRef]
5. Jung, S.; Dorf, R.C. Analytic PIDA controller design technique for a third order system. In Proceedings of the 35th IEEE on Decision and Control, Kobe, Japan, 13 December 1996; Volume 3. [CrossRef]
6. Puangdownreong, D. Application of Current Search to Optimum PIDA Controller Design. *Intell. Control Autom.* **2012**, *3*, 303–312. [CrossRef]
7. Sambariya, D.K.; Paliwal, D. Design of PIDA Controller Using Bat Algorithm for AVR Power System. *Adv. Energy Power* **2016**, *4*, 1–6. [CrossRef]
8. Ukakimaparn, P.; Pannil, P.; Boonchuay, P.; Trisuwannawat, T. PIDA Controller Designed by Kitti's Method. In Proceedings of the ICCAS-SICE, Fukuoka, Japan, 18–21 August 2009; pp. 1547–1550.

9. Smerpitak, K.; Ukakimaparn, P.; Trisuwannawat, T.; La-orsri, P. Discrete-time PIDA Controller designed by Kitti's method with Bilinear transform. In Proceedings of the 12th International Conference on Control, Automation and Systems, JeJu Island, Korea, 17–21 October 2012; pp. 1585–1590.
10. Huba, M. Filtered PIDA Controller for the Double Integrator Plus Dead Time. *IFAC PapersOnLine* **2019**, *52*, 106–113. [CrossRef]
11. Jung, A.; Dorf, R.C. Novel Analytic Technique for PID and PIDA Controller Design. *IFAC Proc. Vol.* **1996**, *29*, 1146–1151. [CrossRef]
12. Sharma, A.; Sharma, H.; Bhargava, A.; Sharma, N. Optimal design of PIDA controller for induction motor using Spider Monkey Optimization algorithm. *Int. J. Metaheuristics* **2016**, *5*, 278–290. [CrossRef]
13. Jitwang, T.; Nawikavatan, A.; Puangdownreong, D. Optimal PIDA Controller Design for Three-Tank Liquid-Level Control System with Model Uncertainty by Cuckoo Search. *Int. J. Circuits Syst. Signal Process.* **2019**, *13*, 60–65, ISSN 1998-4464.
14. Kumar, M.; Hote, Y.V. Robust CDA-PIDA Control Scheme for Load Frequency Control of Interconnected Power Systems. *IFAC PapersOnLine* **2018**, *51*, 616–621. [CrossRef]
15. Mosaad, A.M.; Attia, M.A.; Abdelaziz, A.Y. Whale optimization algorithm to tune PID and PIDA controllers on AVR system. *Ain Shams Eng. J.* **2019**, *10*, 755–767. [CrossRef]
16. Dal-Young, H.; Ihn-Yong, L.; Young-Seung, C.; Young-Do, L.; BooKwi, C. The design of PIDA controller with pre-compensator [for induction motors]. In Proceedings of the 2001 IEEE International Symposium on Industrial Electronics Proceedings (Cat. No.01TH8570), Pusan, Korea, 12–16 June 2001; Volume 2, pp. 798–804. [CrossRef]
17. Raju, M.; Saikia, L.C.; Sinha, N. Automatic generation control of a multi-area system using ant lion optimizer algorithm based PID plus second order derivative controller. *Int. J. Electr. Power Energy Syst.* **2016**, *80*, 52–63. [CrossRef]
18. Guha, D.; Roy, P.K.; Banerjee, S. Multi-verse optimisation: A novel method for solution of load frequency control problem in power system. *IET Gener. Transm. Distrib.* **2017**, *11*, 3601–3611. [CrossRef]
19. Sahib, M.A. A novel optimal PID plus second order derivative controller for AVR system. *Eng. Sci. Technol. Int. J.* **2015**, *18*, 194–206. [CrossRef]
20. Bisták, P.; Huba, M. Analysis of Higher Derivative Degree PID Controllers via Virtual Laboratory. In Proceedings of the 27th Mediterranean Conference on Control and Automation (MED), Akko, Israel, 1–4 July 2019; pp. 256–261. [CrossRef]
21. Lurang, K.; Puangdownreong, D. Two-degree-of-freedom PIDA controllers design optimization for liquid-level system by using modified mat algorithm. *Int. J. Innov. Comput. Inf. Control* **2020**, *16*, 715–732. [CrossRef]
22. Ozbey, N.; Yeroglu, C.; Baykant Alagoz, B.; Herencsar, N.; Kartci, A.; Sotner, R. 2DOF multi-objective optimal tuning of disturbance reject fractional order PIDA controllers according to improved consensus oriented random search method. *Int. J. Innov. Comput. J. Adv. Res.* **2020**, *25*, 159–170. [CrossRef]
23. Huba, M.; Vrančić, D. Comparing filtered PI, PID and PIDD control for the FOTD plants. *IFAC PapersOnLine* **2018**, *51*, 954–959. [CrossRef]
24. Simanenkov, A.L.; Rozhkov, S.A.; Borisova, V.A. An algorithm of optimal settings for PIDD2D3-controllers in ship power plant. In Proceedings of the 2017 IEEE 37th International Conference on Electronics and Nanotechnology (ELNANO), Kiev, Ukraine, 18–20 April 2017; pp. 152–155. [CrossRef]
25. Huba, M.; Vrančić, D. Introduction to the Discrete Time PIDmn Control for the PDT Plant. In Proceedings of the 15th IFAC International Conference on Programmable Devices and Embedded Systems, Ostrava, Czech Republic, 23–25 May 2018.
26. Bistak, P. Disturbance Analysis Virtual Laboratory for PID Controllers with Higher Derivative Degrees. In Proceedings of the 2018 16th International Conference on Emerging eLearning Technologies and Applications (ICETA), Stary Smokovec, Slovakia, 15–16 November 2018; pp. 69–74. [CrossRef]
27. Huba, M.; Vrančić, D.; Bisták, P. PIDmn Control for IPDT Plants. Part 1: Disturbance Response. In Proceedings of the 26th Mediterranean Conference on Control and Automation (MED), Zadar, Croatia, 19–22 June 2018; pp. 1–6. [CrossRef]
28. Skogestad, S. Simple analytic rules for model reduction and PID controller tuning. *J. Process Control* **2003**, *13*, 291–309. [CrossRef]
29. Huba, M.; Vrancic, D. Extending the Model-Based Controller Design to Higher-Order Plant Models and Measurement Noise. *Symmetry* **2021**, *13*, 798. [CrossRef]
30. Vítečkova, M.; Víteček, A. 2DOF PI and PID controllers tuning. In Proceedings of the 9th IFAC Workshop on Time Delay Systems, Prague, Czech Republic, 7–9 June 2010; Volume 9, pp. 343–348.
31. Vrančić, D.; Strmčnik, S.; Hanus, R. Magnitude optimum tuning using non-parametric data in the frequency domain. In Proceedings of the PID'00: Preprints: IFAC Workshop on Digital Control: Past, Present and Future of PID Control, Terrassa, Spain, 5–7 April 2000; pp. 438–443.
32. Vrančić, D.; Strmčnik, S.; Juričić, Đ. A magnitude optimum multiple integration method for filtered PID controller. *Automatica* **2001**, *37*, 1473–1479. [CrossRef]
33. Vrančić, D.; Strmčnik, S.; Kocijan, J.; de Moura Oliveira, P.B. Improving disturbance rejection of PID controllers by means of the magnitude optimum method. *ISA Trans.* **2010**, *49*, 47–56, ISSN 0019-0578. [CrossRef]
34. Vrančić, D. Magnitude optimum techniques for PID controllers. In *Introduction to PID Controllers: Theory, Tuning and Application to Frontiers Areas*; Panda, R.C., Ed.; InTech Cop.: Rijeka, Croatia, 2011; pp. 75–102.
35. Whiteley, A.L. Theory of servo systems, with particular reference to stabilization. *J. IEE Part II* **1946**, *93*, 353–372.
36. Preuss, H.P. Model-free PID-controller design by means of the method of gain optimum. *Automatisierungstechnik* **1991**, *39*, 15–22. (In German) [CrossRef]

37. Kessler, C. Über die Vorausberechnung optimal abgestimmter Regelkreise Teil III. Die optimale Einstellung des Reglers nach dem Betragsoptimum. *Regelungstechnik* **1955**, *3*, 40–49.
38. Åström, K.J.; Panagopoulos, H.; Hägglund, T. Design of PI Controllers based on Non-Convex Optimization. *Automatica* **1998**, *34*, 585–601. [CrossRef]
39. Octave Online Bucket Website. Available online: https://octave-online.net/bucket~{}957TukXTyYuXNMsxeL75FP (accessed on 8 April 2021).