sensors

MDPI

Article

Making the PI and PID Controller Tuning Inspired by Ziegler and Nichols Precise and Reliable

Mikulas Huba [1,*], Stefan Chamraz [1], Pavol Bistak [1] and Damir Vrancic [2]

[1] Institute of Automotive Mechatronics, Faculty of Electrical Engineering and Information Technology, Slovak University of Technology in Bratislava, 812 19 Bratislava, Slovakia; stefan.chamraz@stuba.sk (S.C.); pavol.bistak@stuba.sk (P.B.)

[2] Department of Systems and Control, Jožef Stefan Institute, Jamova Cesta 39, 1000 Ljubljana, Slovenia; damir.vrancic@ijs.si

* Correspondence: mikulas.huba@stuba.sk

Abstract: This paper deals with the design of a DC motor speed control implemented by an embedded controller. The design is simple and brings some important changes to the traditional Ziegler–Nichols tuning. The design also includes a novel anti-windup implementation of the controller and an integrated noise-reduction filter design. The proposed tuning method considers all important aspects of the control, such as pre-processing of the measured signals and filtering (to attenuate the measurement noise), time delays of the process, modeling and identification of the process, and constraints on the control signal. Three important aspects of designing PI and PID controllers for processes with noisy output on Arduino-type embedded computers are considered. First, it deals with the integrated design of the input filter and the controller parameters, since both are interdependent. Secondly, the method of setting the controllers from step responses by Ziegler and Nichols is modified for the case of digital signal processing (without drawing the tangent), while it recommends the suitability of its modification in terms of the use of both integral and static models. Third, the most suitable anti-windup solution for the given controller structure is proposed. In summary, the paper shows that an appropriate design of the embedded controller can achieve excellent closed-loop performance even in a noisy process environment with limited control signals.

Keywords: filtration; multiple real dominant pole method; PI and PID control; derivative action; anti-windup; digitization

check for updates

Citation: Huba, M.; Chamraz, S.; Bistak, P.; Vrancic, D. Making the PI and PID Controller Tuning Inspired by Ziegler and Nichols Precise and Reliable. *Sensors* **2021**, *21*, 6157. https://doi.org/10.3390/s21186157

Academic Editor: Vittorio M.N. Passaro

Received: 17 August 2021
Accepted: 10 September 2021
Published: 14 September 2021

1. Introduction

The introduction of simple single-chip microcomputers has revolutionized process control, which no longer depends on hardware alone. The combination of multiple functionalities in one device makes it possible to bypass the traditional division of the control loop into sensor, controller and actuator. At the same time, the integrated design allows for simplified and accelerated solutions, resulting in increased control performance and robustness.

The concept of PI and PID controllers has been developed in several stages. After their discovery more than a hundred years ago, PI controllers were considered a highly valued commodity, forming a control loop together with sensors and actuators. In the 1930s, a derivative component (called Pre-Act) was added to the PI controllers. The terms proportional-integral or proportional-integral-derivative, from which the abbreviations PI and PID are derived, came later in history. The modular industrial control loops, originally based on pneumatic, hydraulic and electromechanical solutions, changed over time with the advent of transistors, analog integrated circuits and digital technology. The introduction of cheap and powerful embedded computers made it possible to process control algorithms and sensor signals simultaneously. The benefits of integrated design of control algorithms and process signal filtering are so great that they can change the traditional clichés found in control engineering textbooks. The assertion "for noisy systems, it is not appropriate to use controllers with derivative action" [1] is no longer valid and should be changed to "for noisy systems, controllers with derivative action should be used in conjunction with appropriate filtration" [2].

Since its publication, the method for tuning controller parameters by Ziegler and Nichols [3] has been one of the most popular and widely cited, and has inspired many subsequent improvements [4–11]. Originally, the method focused on the best possible suppression of disturbances, to which was also adapted the choice of more "aggressive" settings of controllers, leading mainly to oscillating transients with the so-called "quarter amplitude damping". With the advent of digital computers, the tuning methods were first discretized by introducing the discrete-time transfer functions of the controllers [12], to allow using with longer sampling periods. Later, the numerically sensitive computation of the tangent drawn through the inflection point of the measured step responses was replaced by simpler approximations of the initial segment of the setpoint step responses [13], generalizing the method to higher-order models as well. In this paper, we will focus on digitization yielding two types of linear models obtained from the step responses. The method is further refined to ensure fastest possible transients without overshoot and to allow generalization to higher-order approximations [14].

The systematic design of controllers for noisy systems requires several crucial steps. The first is to appropriately quantify the performance achieved when evaluating transients. The use of traditional statistical tools does not seem appropriate here, since relevant conclusions can only be drawn after a very lengthy investigation (large amount of processed data). On the other hand, the controller must guarantee the required control performance for all considered control processes without requiring a high number of repetitions. Moreover, the noise of the process measurements (sometimes with considerable amplitudes) must be taken into account at the initial stage of the control design, since high peaks in the noise signals could affect the lifetime of the installed electronic devices. The appropriate criteria for evaluating the performance of the achieved process and control transients are based on the early work of Feldbaum [15] in the field of time-optimal control. In [15], it was shown that n control intervals at the process input are generally required for an nth-order system to make a time-optimal transition from one steady state to another, with the control signal alternately changing value between two control signal boundaries. Although Feldbaum's theorem was later forgotten with the advent of linear PID controllers (several alternative approaches to constrained control of time-delayed systems may e.g., be found in [16,17]), it is noted that the theorem is becoming increasingly important as the dynamics of transients increase. This fact was already highlighted in the context of PID control [18] and led to the concept of dynamic classes of control [14,18,19]. Indeed, when designing PID controllers, it is necessary to define the dynamic class of control for which it is proposed.

In the case of a PI controller for a stable first-order system, the ideal course of the control signal following the step of the reference variable to the new constant value may be monotonic, but also with one pulse (1P), i.e., consisting of two monotonic sections separated by an extreme. Instead of a single extreme point, the monotonically increasing/decreasing segments can be separated by a section at a control signal saturation limit. With unstable systems, only the second option is possible. However, the same input and output shapes can be achieved using a PID controller. When designing PID controllers for second-order systems, a third possibility will be added, when the optimal course of the control signal following a setpoint or input disturbance step will consist of two impulses (2P), i.e., three monotonically increasing or decreasing sections separated by two extremes or sections at saturation limits. This means that several types of controllers from the same dynamic class can be designed for the same system, but sometimes several controllers of the same type from different dynamic classes. Examples of controller tuning for second-order systems can be found in [13,20–26]. A broader collection of different approaches gives [27]. Quantitative criteria for evaluating deviations from ideal shapes of transients of process input and output signals have been developed based on deviations from monotonicity [14,28]. In the context of this work, we restrict the design of PI and PID controllers to obtaining a 1P control signal after setpoint and input disturbance steps.

In [3], the relationships for setting the regulators were determined empirically for a wider batch of processes. Given that PID controllers at the time represented a dominant part of the existing supply of industrial regulators, such a procedure could have been natural. However, today's offer, especially if we take into account practically universal programmable embedded controllers, already allows a closer adaptation of the optimal design to a specific type of process. From this point of view, we do not consider the efforts of some newer modifications of the method for universally valid settings [8,9] to be justified. The given trend of narrowing the processes for which the design was optimized can be seen already in the work [12], in which the IPDT system model plays an important role. Given that the IPDT model did not allow a simple explanation of several aspects of the original work of Ziegler and Nichols, in our contributions we also focused on the analysis of other type processes, such as integrator + time constant delay [6,7], double integrator + dead time [13,20,26], or (following the example of the work [29]) approximations with higher-order time constants and transport delay [14]. Instead of numerically ill-conditioned determination of the inflection point and problems with evaluation of steady states, we also preferred the digitization of the method based on the best possible approximation of the introductory part of the measured step response by FOTD and IPDT models. From the closed-loop transients, we expected to reach the highest possible speed under the given control constraints, with the minimal deviations from ideal shapes of piecewise monotonic responses [13,20]. The paper, which is an amalgamation of several publications on the subject (see, e.g., [2,30–48]) provides a refinement of the original step-response-based method of Ziegler and Nichols in terms of the nearly time-optimal responses to step setpoint and input disturbances changes in systems with limited control action also under the influence of measurement noise. It is written pragmatically with the aim of making the subject accessible to the widest possible audience. It is organized as follows: After the basics of system modeling based on approximations of their process response curves by IPDT and FOTD models in Section 2, Section 3 discusses the optimal integrated design of PI and PID controllers with input filters and control signal limitations. Section 4 gives criteria for evaluating the optimality of the closed-loop transients achieved. Section 5 describes an experiment with an electromechanical system. The obtained results are evaluated in Section 6 and summarized in the Conclusions.

2. Possible Approximations of Stable Setpoint Step Responses

The measured process output signal must be filtered before it is further used in the PID controller (see Figure 1) to remove unwanted noise that could reduce the lifespan of the motor and its driver circuitry. Therefore, in addition to the filter, voltage limiters (implemented by diodes) are added to the input of the circuit.

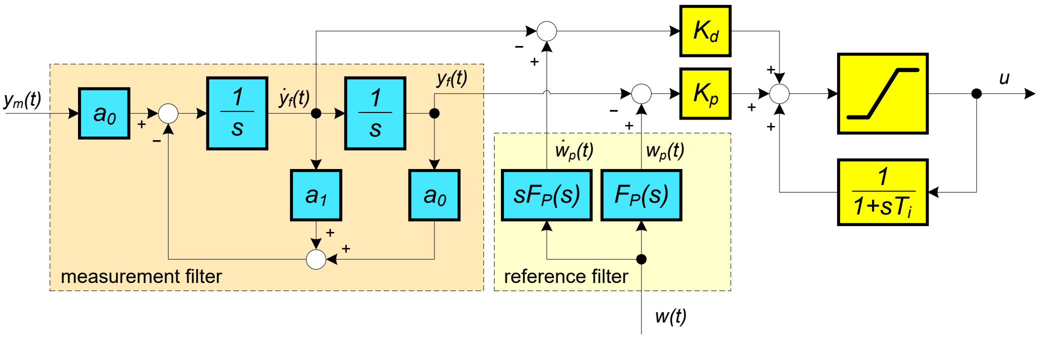

Figure 1. A 2nd-order binomial filter of the measured variable $y_m(t)$ proceeding an anti-windup implementation of PID controller with proportional and derivative gains K_p, K_d, with the integral time constant T_i, with the reference setpoint $w(t)$ and a prefilter $F_p(s)$.

2.1. Input Filter Design

To obtain a filtered velocity signal from the motor ($y_f(t)$) or even its phase vector components $y_f, \dot{y}_f, \ddot{y}_f, \dots y_f^{n-1}$ from raw velocity measurements ($y_m(t)$), a binomial low-pass filter $Q_n(s), n > 1$ is used, which is given by the transfer function

$$Q_n(s) = \frac{Y(s)}{Y_m(s)} = \frac{1}{\left(T_f s + 1\right)^n} = \frac{a_0}{s^n + a_{n-1}s^{n-1} + \cdots + a_1 s + a_0};$$
$$a_k = \frac{n!}{(n-k)!\, k!\, T_f^{n-k}};\ n = 1, 2, \dots\ ;\ k = 0, 1, 2, \dots, n;\ n! = n(n-1)(n-2)\ \dots\ 3*2*1 \tag{1}$$

Please note that $Y_m(s)$ is the Laplace representation of the raw signal $y_m(t)$. The coefficients a_k are calculated by the Pascal's triangle. Please note that for a quasi-continuous-time filter implementation with a sampling period T_s, T_f must satisfy the following condition:

$$T_s << T_f \tag{2}$$

Then, the integration block labeled $1/s$ in Figure 1 can be replaced by a summation performed by a control program implemented in the Arduino environment. The equivalent time delay of the filter can be represented by its average residence time (ART) [1]

$$T_{ART} = nT_f \tag{3}$$

Please note that as the order of the filter n increases, the value of T_f satisfying the condition $T_{ART} = const$ decreases down to the sampling period T_s, which is the fundamental constraint for the choice of n. In the rest of the paper, the filter parameters are calculated according to the process and sensor analysis. To simplify the design of the controller, the identification of the process will include the built-in filter.

2.2. Measuring Input-Output Steady-State Characteristic

When designing controllers for stable systems, it is always important to obtain the steady-state input-output characteristics (IOSSCH) of the process and determine the limits of the system variables. These limits are considered in the identification phase and in the evaluation of the efficiency of the control system. Based on the limits, an appropriate anti-windup protection is also applied. It should be noted that the overall control design also depends on the signal-to-noise ratio.

In the identification phase, the dynamic characteristics of the process are determined. The accuracy of the model is usually evaluated using various quantitative criteria, such as the sum of the squares of the difference between the output of the real system and its model. However, the obtained model is best verified by evaluating the closed-loop transients obtained with the controller designed according to the determined model. As mentioned earlier, the main objective of measuring the IOSSCH of the process is to obtain limit values for the process input and output. The output of the controller is then limited according to the measured process limits. The process limits can be determined in the open loop configuration. If the process is too slow, the closed-loop configuration can be used with a stable control. When the process settles, the pairs of values of process input and process output should be recorded. If the IOSS curve $y_f = F(u)$ is not linear, it is useful to use the inverse characteristic at the controller output to linearize it. Two examples of measuring the IOSSCH together with measured (y_m) and filtered (y_f) output signals are shown in Figure 2.

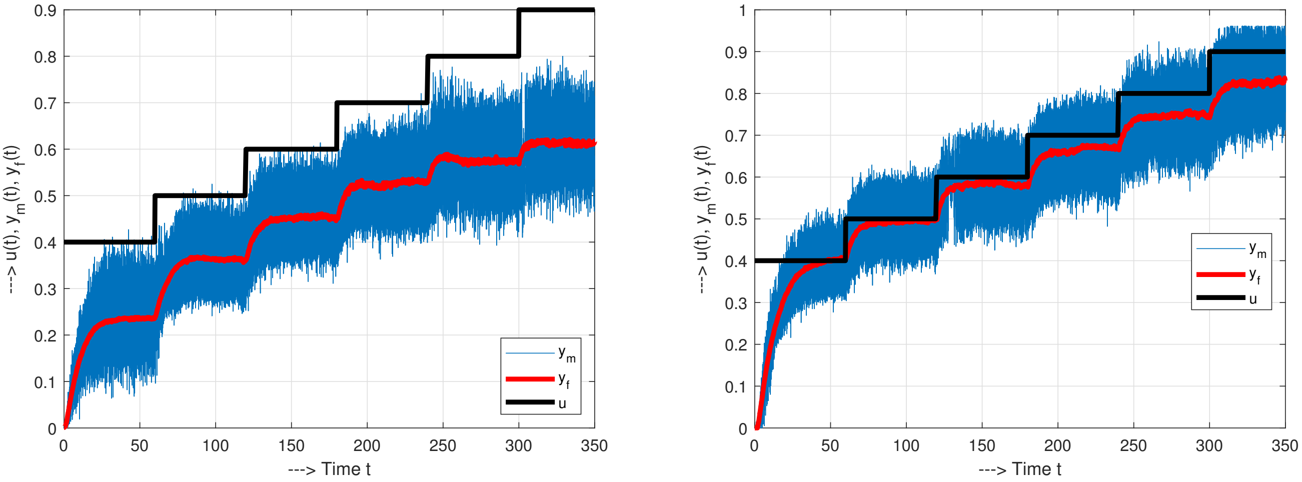

Figure 2. IOSSCHs of a mechatronic system with a measured output y_m, a filtered output y_f and a process input pulse-width u measured before and after maintenance (bearing lubrication).

The use of embedded controllers is often associated with the use of the simplest on-off amplifiers working with pulse-width modulation. In such situations, at a constant power supply amplitude, the pulse width is a variable IOSSCH measurement parameter. Two examples illustrating the measurement of the IOSSCH together with the display of the immediately measured and filtered output signals y_m and y are shown in Figure 2. They document the high time instability of mechatronic devices, which significantly depend on the performed maintenance (bearing lubrication). This instability is one of the motives for the introduction of automatic control and points to the need for a robust control design, which should ensure the minimum required level of control performance independent of such maintenance. As the human factor is also an important element of the design of automatic control, one of the most important goals of the INDUSTRY 4.0 initiative [49,50] is the digitization and automation of the overall control design.

2.3. Approximation of Stable Step Responses According to Ziegler and Nichols

One of the most commonly cited methods for approximating stable step responses, inspired by Ziegler and Nichols [3], can be summarized in the following steps.

First, set an allowable initial value (u_1) of the control variable (plant input) and hold it constant until the transients decay. Measure the corresponding (filtered) value of the output y_{f1}. If the filtered value is still noisy, calculate y_{f1} by averaging several last measurements. Then reset the time $t_0 = 0$ and change the plant input from the value u_1 to the new value u_2 and record the filtered output (graphically or as a series of samples in the computer's memory) until the transients decay (until $t = t_{max}$). Measure the new steady-state value of the output y_{f2} and calculate the static gain of the system K

$$K = \frac{y_{f2} - y_{f1}}{u_2 - u_1} \tag{4}$$

For a chosen input step $\Delta u = u_2 - u_1$, calculate the normalized setpoint response $y_n(t)$ from $y_f(t)$ with time $t \in [0, t_{max}]$ (Figure 3), considering $y(t) = 0$ for $t \le 0$:

$$y_n(t) = \Delta y_f / \Delta u; \; \Delta y_f(t) = y_f(t) - y_f(0); \; t \in [0, t_{max}] \tag{5}$$

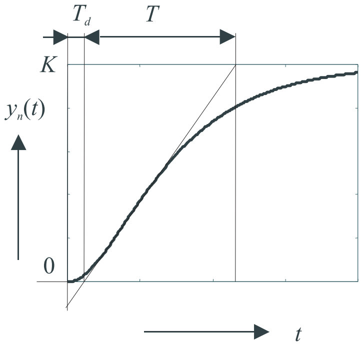

Figure 3. Approximation of a normalized stable step response based on a tangent drawn through the inflation point according to Ziegler and Nichols [3].

From the parameters obtained by drawing the tangent line at the inflection point of $y_n(t)$, one can immediately compute two different linear models of the system. In the conventional linear model, which is locally approximated by the first-order transfer functions with time delay (FOTD)

$$S_a(s) = \frac{\Delta Y(s)}{\Delta U(s)} = \frac{Ke^{-T_d s}}{Ts+1} = \frac{K_s e^{-T_d s}}{s+a}; \tag{6}$$

the three parameters obtained are the transport delay T_d, the time constant T and the static gain of the system K. For $a \neq 0$, the model (6) can also be characterized by

$$K_s = K/T; \; a = 1/T. \tag{7}$$

However, it is possible to construct an even simpler "ultralocal" integrating process model with parameters K_s (which gives the slope of the linearly increasing response given by the tangent line applied to the inflection point) and the dead time T_d

$$S_0(s) = \frac{\Delta Y(s)}{\Delta U(s)} = \frac{K_s e^{-T_d s}}{s}; \tag{8}$$

Here, the third parameter $T = 1/a \to \infty$, which corresponds to $a \to 0$, cannot be evaluated at all.

Remark 1 (Different types of standardized setpoint responses)**.** *If we want to use model* (6)*, we must compute the static gain K and therefore wait until the transients decay. In contrast, model* (8) *can be obtained without determining the static gain, so the experiment can be shorter.*

Remark 2 (Local and ultralocal linear models)**.** *The term "ultralocal" model is used in papers dealing with the design of model-free control using integral models based on the flatness theory elaborated by M. Fliess and colleagues. In previous work [51,52], the two linear (i.e., local) models* (6) *and* (6) *were denoted by the indices 1 or 0 depending on the number of terms in the Taylor series of the considered feedback nonlinearity.*

- The fundamental weakness of both solutions with the model parameterization as shown in Figure 3 is the drawing of the tangent line through the inflection point. This was relatively easy to do with a traditional manual (graphical) solution. However, with numerical solutions, this is one of the most ill-conditioned tasks.
- The second weakness is the instability of the steady states in the system, which are sensitive to various disturbances from the environment. An example of such a step response is shown in Figure 4. The long measurement times can lead to large fluctuations in the process step response due to disturbances. The requirement for

a steady-state measurement without disturbances can be a significant limitation in many applications.

- The third weakness is related to the fact that the obtained models (8) and (6) represent only linear, i.e., local models, which can no longer be sufficiently accurate when a larger range of operating points is applied due to the nonlinearity of the process. Therefore, it is meaningless to require "infinite" measurements to reach the steady state.

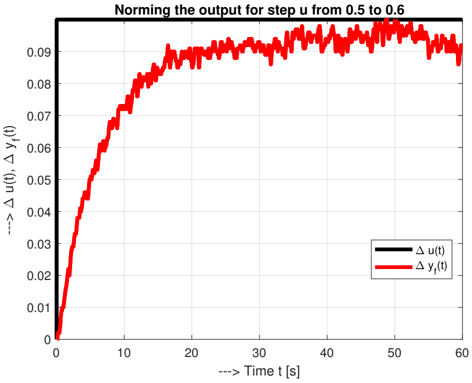

Figure 4. Norming one segment of the step response $y_f(t)$ corresponding to Figure 3 with the change of the input from $u = 0.5$ to $u = 0.6$.

2.4. Local Identification of Local Linear Models

When we model a system, we must always keep in mind the purpose of the model. In our case, the purpose is to design controller that provides a tracking response of the process output that is as fast as possible, but at the same time sufficiently smooth, ideally without overshoot. From this point of view, the process model should have the highest accuracy in the mid to higher frequency range corresponding to the initial response of the process step response in the time domain. The time responses of models (6) and (8) when the step change is $u(t) = 1$ are:

$$y_0(t) = K_s(t - T_d),\ t > T_d;\ \ y_0(t) = 0,\ t \leq T_d \tag{9}$$

$$y_a(t) = \frac{K_s}{a}(1 - e^{-(t-T_d)/T}),\ t > T_d;\ \ y_0(t) = 0,\ t \leq T_d \tag{10}$$

In the following time interval:

$$t \in [0, t_a];\ T_d < t_a \leq t_{max} \tag{11}$$

the parameters of the models (9) and (10) are computed by searching over a grid of predefined values (defined, e.g., in Matlab as $Ks = Ksmin : dK : Ksmax, Td = Tdmin : dT : Tdmax, a = amin : da : amax$) such that the corresponding squares of the deviations

$$S_0 = \sum_{i=0}^{N}(y_{ni} - y_{0i})^2;\ S_1 = \sum_{i=0}^{N}(y_{ni} - y_{ai})^2 \tag{12}$$

take the smallest possible values [13]. In (12), the values $y_{ni} = y_n(iT_s), y_{0i} = y_0(iT_s)$ and $y_{ai} = y_a(iT_s)$ correspond to the measured y_n, y_0 and y_a at the sampling interval T_s.

Therefore, in addition to selecting the appropriate model parameters grid size and fragmentation, the time interval of the process response t_a used for model identification should also be selected. It is obvious that $t_a > T_d$. No control will occur until T_d expires, so you need to prepare for the worst-case scenario. Therefore, when searching for parameters for models (6) and (8) at different values of t_a, the models with the highest value of T_d should be chosen. Such an approach takes into account that the value of T_d plays a crucial role in tuning the controller parameters to achieve smooth transients without overshoot. For the process model (6) (based on $y_a(t)$ in the time domain), higher values of t_a are generally expected to be used. However, if t_a and the process noise are too large (as in the Figure 5b), regarding the accuracy of determining the fastest mode, the obtained model may not be accurate enough.

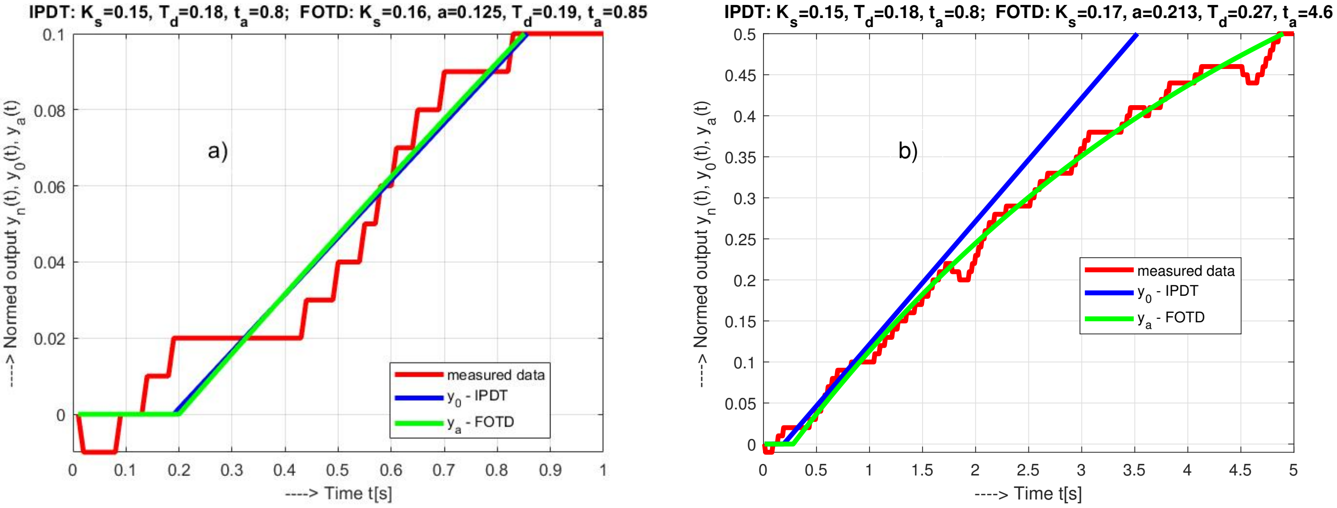

Figure 5. Approximations of an initial part of a normed step response $y(t)$ for $t_a \leq 1$ (**a**) and $t_a \leq 5$ (**b**) with model parameters corresponding to the maximal values T_d from particular "optimal" solutions fulfilling conditions (12).

Remark 3 (Approximating the fastest mode)**.** *Regarding the accuracy of determining the fastest mode, we can formulate a hypothesis that the approximation length t_a should be chosen so that the optimal parameters K_s and T_d are similar for both $S_a(s)$ (6) and $S_0(s)$ (8).*

3. PI and PID Controller Tuning by the MRDP Method

3.1. 2DOF PI and PID Controllers

Although PID controllers are an industry standard, the possible implementation variants have not been sufficiently explored. There are at least two forms (structures) of PID controllers.

One of them is an ideal series PID controller:

$$C(s) = \frac{U(s)}{E(s)} = K_p \frac{(1+sT_i)(1+T_D s)}{sT_i} = K_p \left(1 + \frac{1}{T_i s}\right)(1 + T_D s) \tag{13}$$

where K_p is the controller gain, T_i is the integral, and T_D is the derivative time constant. The second form is an ideal parallel PID controller:

$$C(s) = \frac{U(s)}{E(s)} = K_p \frac{1 + sT_i(1+T_D s)}{sT_i} = K_p \left(1 + \frac{1}{T_i s} + T_D s\right) \tag{14}$$

Please note that the parallel form is more general, since any series structure can be represented by the parallel structure, while the converse is not true. Please note that the optimal controller parameters for one structure are not optimal for the other (the parameters must be recalculated). As soon as we use both controllers at the same time, their parameters will need to be clearly distinguished by a suitable choice of symbols (see Section 3.4). The PI controllers are simplified PID controllers corresponding to $T_D = 0$. The 2DOF PID or PI controllers represent an extension of a PID controller $C(s)$ with one degree of freedom (1DOF) by a setpoint prefilter $F_p(s)$. The prefilter is added to speed up setpoint responses and avoid overshoot of the process output.

The anti-windup solution for the series PID controller is realized with the structure modeling pneumatic regulators, which were created before the Second World War with the original name “Pre-Act”. The integral action is accomplished by positive feedback from the controller output through the first-order filter with a time constant T_i (see Figure 1 and note that $K_d = K_p T_D$). If the limitations of the control action are neglected, we obtain a combination of PD controller $K_p(1 + T_D s)$ and positive feedback with a delay of $1/(1 + T_i s)$, yielding the transfer function (13)

$$C(s) = \frac{U(s)}{E(s)} = \frac{K_p(1 + T_D s)}{1 - 1/(1 + sT_i)} = K_p \frac{(1 + T_i s)(1 + T_D s)}{T_i s} \tag{15}$$

In the analysis of anti-windup controllers (see e.g., [53,54]), it is necessary to distinguish two modes of behavior—“control” mode, when the limitation is not active and the dynamics is determined by the transfer function $C(s)$, and “tracking” mode, which is activated after reaching saturation limits, when the controller state is tracked so that it matches given inputs and outputs. With the scheme used, the tracking time constant $T_t = T_i$, which according to [53] represents an unnecessarily large value, leading to the output overshooting. However, the overshooting can be avoided if, in the factorization of the numerator $C(s)$ discussed in Section 3.4, we choose as T_i the smaller one of two possible values.

3.2. Optimal PI Tuning by TRDP Method

The goal of the Multiple Real Dominant Poles (MRDP) method is to avoid slow dominant closed-loop poles by choosing equal multiple closed-loop poles (see, e.g., [23,55–57] and references therein). The existence of an MRDP solution must be supported by a suitable controller structure and a sufficient number of equations to determine the controller parameters. If there are several possible solutions with multiple dominant poles, the most stable solution is chosen, i.e., the solution that lies in the left complex half-plane closest to the imaginary axis. The transfer function of the closed loop using the PI controller and the FOTD system (6) is

$$F_{cl}(s) = \frac{C(s)S(s)}{1 + C(s)S(s)} = \frac{K_p K_s(1 + T_i s)}{T_i s(s + a)e^{T_d s} + K_p K_s(T_i s + 1)} \tag{16}$$

The “optimal” controller tuning must achieve a triple real dominant pole (TRDP) of the characteristic quasi-polynomial:

$$P(s) = T_i s(s + a)e^{T_d s} + K_p K_s(T_i s + 1) \tag{17}$$

when it is possible to write $P(s) = (s - s_o)^3 P_0(s)$, by satisfying the conditions

$$P(s_o) = 0,\ \dot{P}(s_o) = 0,\ \ddot{P}(s_o) = 0 \tag{18}$$

The optimal controller parameters, denoted by the subscript o, can be calculated as follows [30]

$$\begin{aligned} s_o &= -\frac{A_d + 4 - S_d}{2T_d}\ ,\quad A_d = aT_d\ ,\quad S_d = \sqrt{A_d^2 + 8} \\ K_{po} &= \frac{K_o}{K_s T_d};\ K_o = (S_d - 2)e^{(S_d - A_d - 4)/2} \\ T_{io} &= \tau_i T_d;\ \tau_i = \frac{2(2 - S_d)}{A_d^2 + 2A_d + 28 - (A_d + 10)S_d} \end{aligned} \tag{19}$$

Denominator of an optimal prefilter

$$F_p(s) = \frac{1 + bs}{1 + T_i s} \tag{20}$$

cancels the zero of the closed-loop transfer function (16). Then, the transients of the closed loop can be sped up by cancelling one of the optimal dominant poles (s_o). This results in its optimum tuning b_o as

$$b_o = -\frac{1}{s_o} = \frac{2T_d}{A_d + 4 - S_d} \tag{21}$$

For the IPDT model (8), much simpler tuning formulas can be derived since $a = 0$ [55,58] and

$$\begin{aligned} &s_o = -(2-\sqrt{2})/T_d \approx 0.586/T_d; \\ &K_o = 2(\sqrt{2}-1)e^{\sqrt{2}-2} \approx 0.461;\ \tau_i = (2\sqrt{2}+3) \approx 5.828;\ b_o = \frac{T_d}{2-\sqrt{2}} \approx 1.707T_d \end{aligned} \tag{22}$$

3.3. Calculation of the Optimal Parallel PID Parameters

The optimal parallel PID setting for a FOTD system, which results in a quadruple real dominant pole (QRDP) in the closed-loop system, is described in [31]. Considering the closed-loop system

$$F_{cl}(s) = \frac{C(s)S(s)}{1 + C(s)S(s)} = \frac{K_p K_s (1 + T_i s(1 + T_D s))}{T_i s(s+a)e^{T_d s} + K_p K_s (1 + T_i s(1 + T_D s))} \tag{23}$$

the characteristic quasi-polynomial is

$$P(s) = T_i s e^{T_d s}(s + a) + K_p K_s (T_i T_D s^2 + T_i s + 1) \tag{24}$$

The "optimal" parameters K_{po}, T_{Do} and T_{io} satisfying the conditions $P(s_o) = 0$, $\dot{P}(s_o) = 0$, $\ddot{P}(s_o) = 0$ and $\dddot{P}(s_o) = 0$, are given as

$$\begin{aligned} &s_o = -\frac{6 + A - S}{2T_d}\ ,\quad A = aT_d\ ,\quad S = \sqrt{A^2 + 12} \\ &K_{po} = \frac{K_o}{K_s T_d};\ K_o = \frac{S(A+12) - (A^2 + 2A + 36)}{2} e^{(S-A-6)/2} \\ &T_{Do} = \tau_D T_d;\ \tau_D = \frac{S-2}{S(A+12) - (A^2 + 2A + 36)} \\ &K_{Do} = K_{po} T_{Do} = \frac{S-2}{2K_s} e^{(S-A-6)/2} \\ &T_{io} = \tau_i T_d;\ \tau_i = \frac{2(36 + 2A + A^2 - (A+12)S)}{A^3 + 12A^2 + 36A + 288 - (A^2 + 12A + 84)S} \end{aligned} \tag{25}$$

3.4. Calculation of Optimal Series PID Parameters

For the series PID controller, the optimal QRDP position s_o is the same as for the parallel controller (25). To avoid tedious calculations and to highlight the main difference in tuning the parallel and series controllers, we obtain the parameters of the series PID controller from the parallel controller by equating the transfer functions $C(s)$ of both controllers. By comparing the coefficients at equal powers of 's' in (13) and (14) and using the indices "*s*" and "*p*", we can derive the following relations.

$$K_{ps} T_{ip} = K_{pp} T_{is};\ T_{is} + T_{Ds} = T_{ip};\ T_{is} T_{Ds} = T_{ip} T_{Dp} \tag{26}$$

Based on the solution of the last two equations with respect to T_{is}, the following is obtained

$$\begin{aligned} &T_{is} = [T_{ip} \pm \sqrt{(T_{ip}^2 - 4T_{ip}T_{Dp})}]/2 \\ &K_{ps} = K_{pp}[T_{ip} \pm \sqrt{(T_{ip}^2 - 4T_{ip}T_{Dp})}]/(2T_{ip}) = K_{pp}[0.5 \pm \sqrt{(0.25 - T_{Dp}/T_{ip})}] \\ &T_{Ds} = T_{ip} \mp \sqrt{(T_{ip}^2 - 4T_{ip}T_{Dp})}]/2 \end{aligned} \tag{27}$$

The expressions (27) show that there are two optimal sets of parameters for the series PID controller. We have already pointed out this ambiguity in [44], and the conclusion was that it has no practical impact when the system is linear. However, it turns out to be important for the design of an anti-windup PID structure with integral action generated by a positive feedback from the output of the constrained controller with time constant T_i, as shown in Figure 1 (all parameters of the controller correspond to the series PID). On the other hand, the optimal setting of the series PID controller is much simpler using the IPDT model (the subscript "s" is omitted for simplicity). The MRDP method yields for (8)

$$\begin{aligned} &s_o = (\sqrt{3} - 3)/T_d;\ K_o = K_{po}K_sT_d = 0.7239 \\ &\tau_i = T_{io}/T_d = 3.4475;\ \tau_D = T_{Do}/T_d = 0.284 \end{aligned} \tag{28}$$

Let us now consider an ideal series $C_a(s)$ controller

$$\begin{aligned} C_a(s) &= K_{pa}\frac{(1 + T_a s)(1 + T_b s)}{T_a s} = \\ &= K_{pa}\left(1 + \frac{1}{T_a s}\right)(1 + T_b s) => T_i = T_a;\ T_D = T_b \end{aligned} \tag{29}$$

with parameters K_p, T_a, T_b satisfying the closed-loop quadruple real dominant pole condition, where T_a has been interpreted as T_i and T_b as T_D. This controller is obviously equivalent to the controller

$$\begin{aligned} &C_b(s) = K_{pb}\left(1 + \frac{1}{T_b s}\right)(1 + T_a s) \\ &K_{pb} = K_{pa}\frac{T_b}{T_a};\ \ => T_i = T_b;\ \ T_D = T_a \end{aligned} \tag{30}$$

Although the first set of parameters (28) was chosen to yield values for τ_i and K_o that are close to PI control (22), the coefficients in (30) yield the second alternative with significantly different parameters ($K_o = 0.0596$). The importance of this ambiguity can be shown in the case of constrained control (with the controller output limitations).

3.5. Prefilter Calculation

When changing from a parallel to a series PID controller, it is also necessary to change the calculation of the optimal prefilter, which is assumed as follows

$$F_p(s) = \frac{1 + bs + cs^2}{(1 + T_i s)(1 + T_D s)} \tag{31}$$

In the simplest case, it is designed to compensate for the zeros of the closed-loop transfer function

$$F_{cl}(s) = \frac{C(s)S(s)}{1 + C(s)S(s)} = \frac{K_pK_s(1 + T_i s)(1 + T_D s)}{T_i s(s + a)e^{T_d s} + K_pK_s(1 + T_i s)(1 + T_D s)} \tag{32}$$

With the trivially chosen setpoint weights

$$b_0 = c_0 = 0 \tag{33}$$

the entire control transfer function becomes an I-PD controller. In the general case, the prefilter parameters can be derived from the polynomial equations

$$\begin{aligned} N_1(s) &= (1 - s/s_o) = 1 + \frac{2T_d}{6 + A - S}s\ , \quad A = aT_d\ , \quad S = \sqrt{A^2 + 12} \\ N_2(s) &= (1 - s/s_o)^2 = \left(1 + \frac{2T_d}{6 + a - S}s\right)^2 \end{aligned} \tag{34}$$

to cancel out one or two dominant poles s_o. For (8) the result is

$$\begin{aligned} b_1 &= 0.7887T_d;\ c_1 = 0; \\ b_2 &= 1.5774T_d;\ c_2 = 0.6220T_d^2. \end{aligned} \tag{35}$$

In the experimental evaluation, we preferred to maintain the continuity of the course of the reference signal and so we worked only with $c = 0$.

4. Time and Shape-Related Performance Measures

The integral of absolute error (IAE) is often used to evaluate the speed of closed-loop response:

$$IAE = \int_0^\infty |e(t)|dt\ ;\ \ e = w - y\ . \tag{36}$$

Since the controller optimization based on the IAE criterion leads to overshoot of the output in most cases, additional constraints must be applied to achieve a monotonic response. In this work, the additional optimization constraints are considered in the form of shape-related constraints [2,14]. These are based on the concept of monotonicity, where the monotonicity measure is proposed as

$$TV_0(y) = \sum_{i=0}^{\infty} |y_{i+1} - y_i| - |y_\infty - y_0| \tag{37}$$

i.e., as the difference between the total signal path (y) (sum of increments $|y_{i+1} - y_i|$) and the minimum possible path required to change the signal from the initial state (y_0) to the final state (y_∞). For monotonic responses (with no overshoot or undershoot of $y(t)$), $TV_0(y) = 0$. Otherwise, $TV_0(y) > 0$.

The evaluation of signal shapes based on deviations from monotonicity can be viewed as a generalization of the evaluation approaches used in the early stages of automatic control in relay time-optimal control. In this context, references to Feldbaum's theorem [14,15,59,60] can be found in the older literature. According to this theorem, to move the output of an nth-order system (with full relative order) from one steady state to another, n control intervals are required in which a controlled variable changes its values from one limit to the opposite. Such a rectangular pulse can be considered to be a limiting case of two monotonically increasing/decreasing responses separated by an interval with a constant value of the constraint. For example, a simple integrator requires an input signal consisting of two monotonic intervals for a monotonic output transition of the steady state [58].

To evaluate deviations from such an ideal single-pulse waveform (1P) [58] consisting of two monotonic intervals separated by an extreme value $u_m \notin (u_0, u_\infty)$ that lies outside the interval of the initial and final output values u_0 and u_∞, the measure $TV_1(u)$

$$TV_1(u) = \sum_i |u_{i+1} - u_i| - |2u_m - y_\infty - u_0| \tag{38}$$

can be used. To separate the measured deviations due to the change of the steady state from the deviations caused by the process noise, the evaluation interval can be restricted to the time at which the process response practically settles ($|e(t)| \leq e_0$, where e_0 is chosen according to the noise level). Simpler relationships for the IPDT model can be used to demonstrate the effect of T_d on the rate of transients. For example, in the PID control yielding ideal shapes (with zero deviations at the plant output), the optimal IAE values characterizing unit setpoint step responses can be calculated using the Laplace transform [44] as integral of error (IE) values, when

$$IAE = T_i + T_D - b \tag{39}$$

With the simplest prefilters (33) and (35)

$$IAE_0 = 3.732T_d;\; IAE_1 = 2.9433T_d;\; IAE_2 = 2.1546T_d \tag{40}$$

The results of the analysis of the influence of T_d on the speed of transients in the design using MRDP method make it possible to formulate the following note.

Remark 4 (Motivation to search for models with a maximum dead-time value)**.** *If we used a smaller T_d than the actual one when setting the controller, the transients would be accelerated. However, since the TRDP corresponds to a limit setting at which oscillations do not yet occur, this would be at the cost of disrupting the ideal waveforms. To avoid this, we look for the IPDT model with the largest T_d value when identifying the plant.*

5. Illustrative Example

Mechatronics is an initiative started in Japan around 1970 that aims to improve the functionality of various products by making them more accurate, flexible, adaptable, or user-friendly - all with the goal of reducing production costs. With similar goals in mind and to test the dynamics of electric vehicle models, in 1982 we built a low-cost speed control system with three commutator DC motors. The mechanical construction of the vehicle model was made using the Meccano kit from Merkur (Figure 6). At that time, the device was controlled by a HP-85 computer, which allowed the use of self-adjusting and adaptive controllers. The sampling time was between 0.1 and 0.3 s, which is reasonable given the main time constant of the system (about 5 s).

From a mechanical point of view, the car model was not very close to the actual car, but it was suitable for testing and improving the self-adjusting and adaptive control algorithms mentioned above, including the study of nonlinear and constrained systems, time-varying systems, measurement and quantization noise, resonances, etc., which were later applied to various systems. The advent of embedded computers with much higher computational power in recent decades enabled the experimentation of various control approaches, which were previously not possible due to the limited computational power of microprocessors. Here, the aforementioned electromechanical system proved to be a suitable candidate to demonstrate recent advances in control approaches and to show that many control shortcomings of the past can be easily addressed today.

The device consists of a DC motor driven by a pulse-width modulated (PWM) motor driver that converts the process input voltage into 7V high pulses of the same frequency and varying duty cycle. The displayed process input value u is normalized between 0 and 1, where 1 means a duty cycle of 100% or constant 7 V and 10 ms at the motors. The process output is the rotational speed of the system measured by the third DC motor. To attenuate the significant measurement noise (see Figure 2), highlighted due to the use of a simple motor instead of a special tachodynamo, the 2nd order filter in Figure 1 was extended to the 4th order binomial filter (1) used with time constant $T_f = 50$ ms at sampling period $T_s = 0.1$ ms, which satisfies condition (2). It should be borne in mind here that the sampling period T_s cannot be reduced arbitrarily, even if the signal processing speed would allow it. As T_s decreases, also useful output changes per sampling period decrease, increasing the effect of quantization noise. At the selected T_s value, the effect of output level quantization against measurement noise was still negligible. The sampling period used for the control algorithm was $T_s = 10$ ms. For all measurements, the output speed of the process is normalized to the fastest speed, which is 13.7 revolutions per second (RPS). Therefore, the indicated speed in the diagrams is between 0 and 1. Experiments with the built device complement similar research carried out with a speed system consisting of a DC motor with an incremental speed sensor [61]. Thanks to very simple construction enabling the work with a higher moment of inertia and a larger mechanical time constant (and an easy verification of the presented research by the readers), the sampling period can be chosen in a wide range. With the chosen experiment setup, we do not want to question the advantages of incremental sensors, but to explore the possibilities of improving the noisy output from analog sensors, which are still important in a high number of practical applications. The sensing of the speed by the commutator tachodynamo leads to the examination of a substantially different situation with a different character of the acting noises. The source of the signal fluctuation is mainly the rotation of the rotor winding in a variable magnetic flux and the switching of individual coils. By shortening the sampling period and increasing the inertia of the filters, we obtain a roughly constant speed signal from the measured signal at a constant rotational speed. The experimental results fully confirm these expectations.

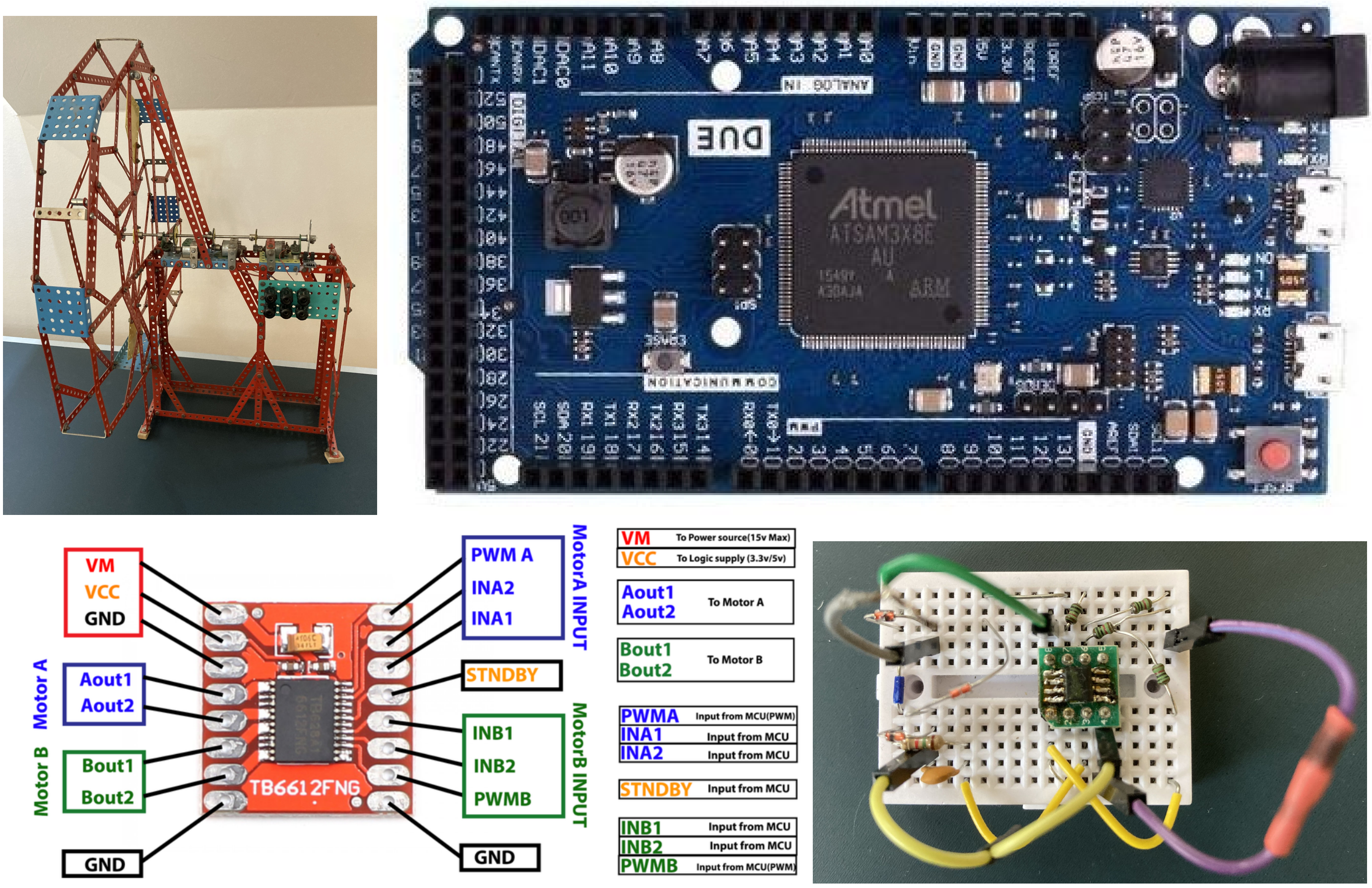

Figure 6. Components of the considered electromechanical Arduino-Due-based speed system.

When evaluating the individual step responses in Figure 2, it is clear that the system is nonlinear and time-varying: After prolonged experimentation, the motor characteristics change with time due to the increased motor temperature. They also change significantly after each lubrication of the bearings. The variations in velocity in the vicinity of steady states are mainly due to imperfect mechanics and resonances that occur. This means that the steady-state identification may not be accurate enough and repeatable to achieve a reliable design of the desired closed-loop dynamics. Therefore, the classical identification methods introduced by Ziegler and Nichols [3] should be modified.

5.1. Specifying the Local and Ultralocal Linear Process Approximations

The analysis and modifications of Ziegler and Nichols' method have been a constant since the 1940s. Let us mention here at least the modifications concerning the digitization of the model transfer function [12]. Here we will further develop the model specifications according to [13]. However, the actual list may be more extensive as new approaches are developed regularly. The basic motivation for the proposed modifications is to increase the reliability in real life (applications) and to obtain more efficient closed-loop transients. We add to them the need for the widest possible digitization of the overall design. To achieve this, the new methods of controller design and the new criteria for evaluating the performance of transients are used. However, the fundamental problem of these methods is the reliable determination of the model parameters (in our case (6) and (8)). The design of a controller can be concise only if the model used in its design approximates with sufficient accuracy the fastest components of the transients, which corresponds to an accurate estimation of the process in the higher frequency range. If we accurately approximate slow components with it (i.e., if we focus on estimation of the process model at low frequencies), the design can be completely degraded by the behavior of fast components. The fact that a model used excellently approximates the measured process reaction curve over a wide time, can be irrelevant for a fast and smooth setpoint tracking.

The models in Figure 5a were obtained by searching for optimal parameters on the grid defined by the intervals:

$$\begin{aligned}
&Td \in [0, 50T_s],\ \Delta T_d = T_s;\\
&K_s \in [0.1, 1.5],\ \Delta K_s = 0.01;\\
&T \in [2, 8],\ \Delta T = 0.1.
\end{aligned} \tag{41}$$

In the presence of measurement noise, the basic way to refine the model parameters is to extend the evaluated interval. However, this does not apply indefinitely. For example, consider an identification interval specified by $t \in [0, t_a]$. Since optimal results cannot be expected for every identification interval, consider several approximations with $t_a \in [0.5, 1]$, all of which have specific parameter values (41). The optimal setting in each case must satisfy conditions (12). The global optimum can be sought under the condition of the maximum value of T_d, since the dead time is usually the most limiting parameter of the controller design. Figure 5a shows that for $t_a \leq 1$, the responses of both models are practically the same. Therefore, the values of K_s and T_d of the two transfer functions (6)–(8) considered are approximately the same:

$$\begin{aligned}
&\text{IPDT},\ t_a \leq 1:\ K_s = 0.15;\ T_d = 0.18;\ a = 0;\ t_{ao} = 0.8\\
&\text{FOTD1},\ t_a \leq 1:\ K_s = 0.16;\ T_d = 0.19;\ a = 0.125;\ T = 1/a = 8;\ t_{ao} = 0.85
\end{aligned} \tag{42}$$

Here t_{ao} denotes the identification interval that yields the optimal parameter set. If we extend the possible length of the identification to $t_a \in [0.5, 5]$, we obtain

$$\begin{aligned}
&\text{IPDT},\ t_a \leq 5:\ K_s = 0.15;\ T_d = 0.18;\ a = 0;\ t_{ao} = 0.8\\
&\text{FOTD5},\ t_a \leq 5:\ K_s = 0.17;\ T_d = 0.27;\ a = 0.213;\ T = 1/a = 4.7;\ t_{ao} = 4.6
\end{aligned} \tag{43}$$

The IPDT model has not changed, but the FOTD model, which estimates the process response more accurately, leads to significantly larger values of all determining parameters than for $t_a \in [0.5, 1]$.

5.2. Experimenting with PI Control

By choosing the identification interval $t_a \leq 1$ the parameters of the PI controllers (19)–(22) are

$$\begin{aligned} PI_0: \ K_{Po} &= 17.07995526; \ T_{io} = 1.049116873; \ b_o = 0.3072792204 \\ PI_{a1}: \ K_{Po} &= 14.99317409; \ T_{io} = 1.034359438; \ b_o = 0.3179322586 \end{aligned} \tag{44}$$

Comparing the controller parameters obtained from both models, it can be seen that the biggest difference is in the gains K_{Po}. Repeating the identification in a larger interval $t_a \leq 5$, the parameters corresponding to FOTD5 are as follows

$$PI_{a5}: \ K_{Po} = 9.771989345; \ T_{io} = 1.338369226; \ b_o = 0.4395610608 \tag{45}$$

In this case, the controller parameters obtained from the IPDT and FOTD models differ significantly.

Figures 7 and 8 show the setpoint responses of the PI controllers designed according to the two considered models and identification intervals. A slight overshoot of the output responses corresponding to the FOTD model with $t_a \leq 1$ is a consequence of the limitation of the control signal and the nonlinearity of the process. Indeed, the overshoot does not occur for smaller setpoint changes. However, it should be noted that the overshoot would be much larger without anti-windup protection.

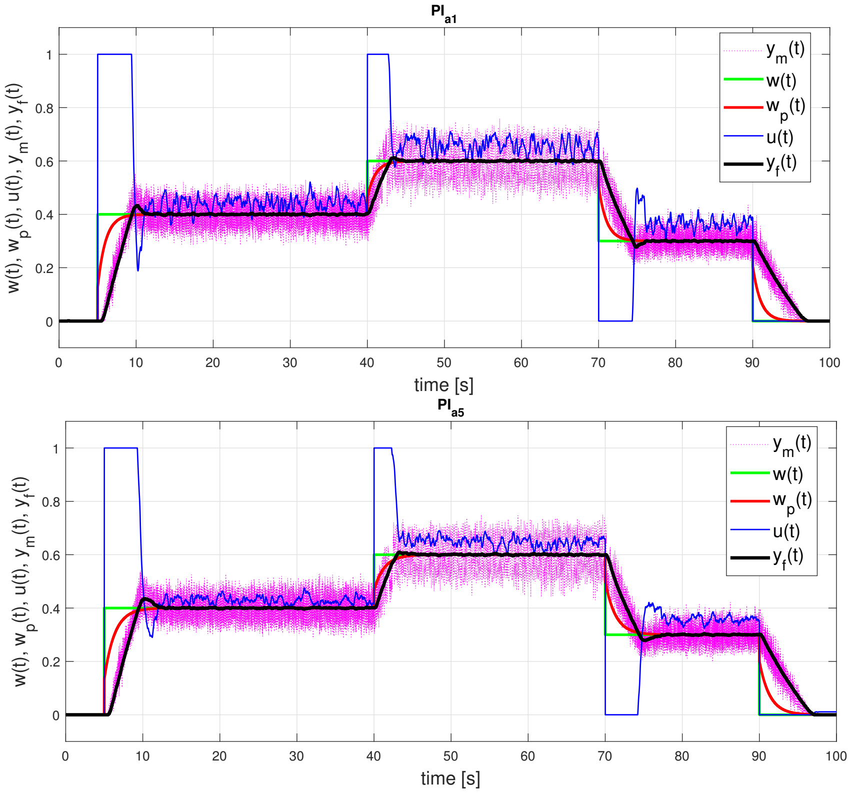

Figure 7. PI control: setpoint responses corresponding to FOTD approximations of an initial part of a normed step response $y_n(t)$ for $t_a \leq 1$ (42) with the parameters (44) (above) and for $t_a \leq 5$ (43) with the parameters (45) (below).

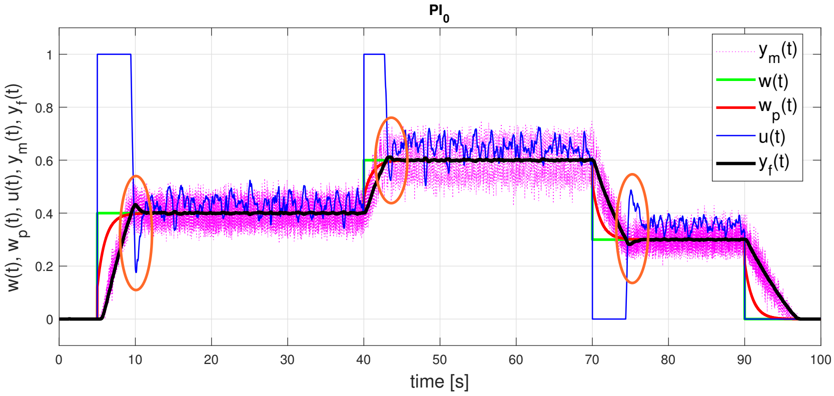

Figure 8. PI_0 control: setpoint responses corresponding to IPDT approximations of an initial part of a normed step response $y_n(t)$ (42) with the parameters (44) and with emphasis on areas including overshooting output requiring compensation by an additional input pulse.

In the case of the FOTD model over a longer identification interval $t_a \leq 5$, a lower controller gain leads to lower $TV_1(u)$ values (see Table 1). A small overshoot of the output occurs in both process models used. It is obvious that neither the transients nor the performance measures are better when a more complex FOTD model is used (for both $t_a \leq 1$ and $t_a \leq 5$).

5.3. Experimentation with PID Control

The calculated PID control parameters (19)–(22) for $t_a \leq 1$ are

$$\begin{aligned}
&PID_{10}: \\
&K_{Po} = 26.80948841;\; T_{io} = 0.6205422427;\; T_{Do} = 0.05122690297;\; b_o = 0.1419615242 \\
&PID_{20}: \\
&K_{Po} = 2.213172556;\; T_{io} = 0.05122690297;\; T_{Do} = 0.6205422427;\; b_o = 0.1419615242 \\
&PID_{1a1}: \\
&K_{Po} = 23.61125885;\; T_{io} = 0.6289503085;\; T_{Do} = 0.05389188106;\; b_o = 0.1484626127 \\
&PID_{2a1}: \\
&K_{Po} = 2.023140996;\; T_{io} = 0.05389188106;\; T_{Do} = 0.6289503085;\; b_o = 0.1484626127
\end{aligned} \tag{46}$$

The largest differences in the controller parameters exist for the gain K_{Po} in both models. Performance measures of the responses corresponding to the FOTD model in Table 1 show that despite the large differences in the controller parameters, the differences in the closed-loop responses are relatively small. The responses corresponding to the IPDT model in Figure 9 are very close to the responses obtained with a more complex FOTD model. However, a closer look at the responses reveals that we obtain monotonic (and nearly time-optimal) process output responses and 1P control signal responses only for the second set of controller parameters with a smaller value of K_p. The control loop responses corresponding to the first set of parameters exhibit small overshoots of the process output responses (increased PO values in Table 1 and slightly increased $TV_0(y)$ values) and the second impulse on the controller output response.

Table 1. Performance measures IAE, $TV_0(y)$ and $TV_1(u)$ corresponding to three steps for all experiments; PO—percentage overshoot; minimum—blue; maximum—red.

	Step from 0 to 0.4				**Step from 0.4 to 0.6**				**Step from 0.6 to 0.3**			
	IAE	$TV_0(y)$	$TV_1(u)$	PO	IAE	$TV_0(y)$	$TV_1(u)$	PO	IAE	$TV_0(y)$	$TV_1(u)$	PO
PI_0	**1.023**	0.096	1.154	8.00	0.314	**0.110**	1.678	6.00	**0.656**	**0.066**	0.870	6.00
PI_{a1}	1.043	0.106	1.158	8.25	0.312	0.094	1.296	6.00	0.667	**0.092**	1.096	**7.67**
PI_{a5}	1.064	0.098	**0.584**	**8.75**	**0.311**	0.088	**0.744**	**5.00**	0.679	0.072	**0.480**	7.33
PI_{a20}	**1.127**	**0.086**	0.918	8.00	0.316	**0.064**	0.972	6.00	0.710	0.074	0.874	6.33
PI_{a20X}	1.078	**0.118**	**3.000**	**6.00**	**0.324**	0.102	**3.060**	**8.00**	**0.729**	0.080	**2.298**	**4.67**
PID_{10}	1.056	0.080	2.530	5.50	**0.319**	0.126	4.204	6.50	0.693	0.066	2.062	4.67
PID_{20}	1.105	0.056	2.372	1.00	0.342	0.086	3.578	1.50	0.701	**0.032**	1.730	**0.33**
PID_{1a1}	1.061	**0.106**	2.920	**6.25**	0.326	0.106	3.464	5.50	0.681	0.054	1.922	4.33
PID_{1a5}	1.101	0.094	1.646	5.75	0.342	0.094	**2.476**	3.00	0.699	0.056	**1.140**	3.67
PID_{2a1}	1.084	0.078	2.740	1.25	0.334	0.116	3.974	4.00	**0.663**	0.054	2.026	1.00
PID_{2a5}	**1.342**	**0.032**	1.944	**0.25**	**0.583**	0.068	2.620	1.50	**0.886**	0.034	1.834	**0.33**
PID_{1a20}	1.143	0.072	2.066	5.25	0.322	0.076	2.758	6.00	0.703	0.068	2.456	**5.00**
PID_{2a20}	1.168	**0.032**	**1.628**	0.50	0.337	**0.044**	2.572	**1.00**	0.711	0.044	2.170	0.67
PID_{1a20X}	**1.012**	0.104	**3.026**	5.25	0.334	**0.160**	5.438	**7.50**	0.740	**0.078**	**2.468**	4.67
PID_{2a20X}	1.065	0.072	2.776	1.00	0.332	0.144	**5.592**	5.00	0.764	0.042	1.452	0.67

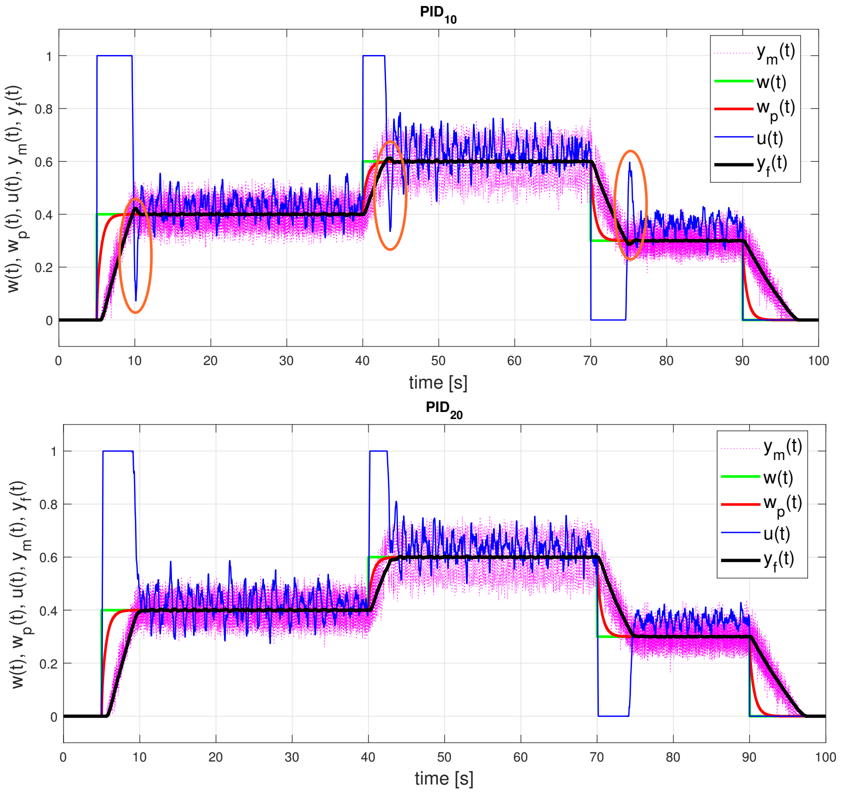

Figure 9. PID control: setpoint responses corresponding to IPDT approximation of an initial part of a normed step response $y_n(t)$ for both possible sets of the controller parameters (46) with emphasis on areas with overshooting output and with excessive input pulse of PID_{10}.

By evaluating the controller parameter for $t_a \leq 5$ we obtain

$$
\begin{aligned}
&PID_{1a5}:\\
&K_{Po} = 13.24788546;\ T_{io} = 1.408562174;\ T_{Do} = 0.07048046952;\ b_o = 0.1484626127\\
&PID_{2a5}:\\
&K_{Po} = 0.6628867394;\ T_{io} = 0.07048046952;\ T_{Do} = 1.408562174;\ b_o = 0.1484626127
\end{aligned}
\tag{47}
$$

In this case, the difference of the controller parameters, except for the setpoint weighting factor b, is again obvious.

As can be seen in Figure 10, the differences in the control responses are relatively small despite the large differences in the controller parameters. The response of the PID_{a52} controller is slightly less overshooting than that of PID_{a51}. The responses with PID_{a52} are already slightly overdamped—the control signal does not go into saturation during the second and third setpoint changes. As a result, the IAE values increase.

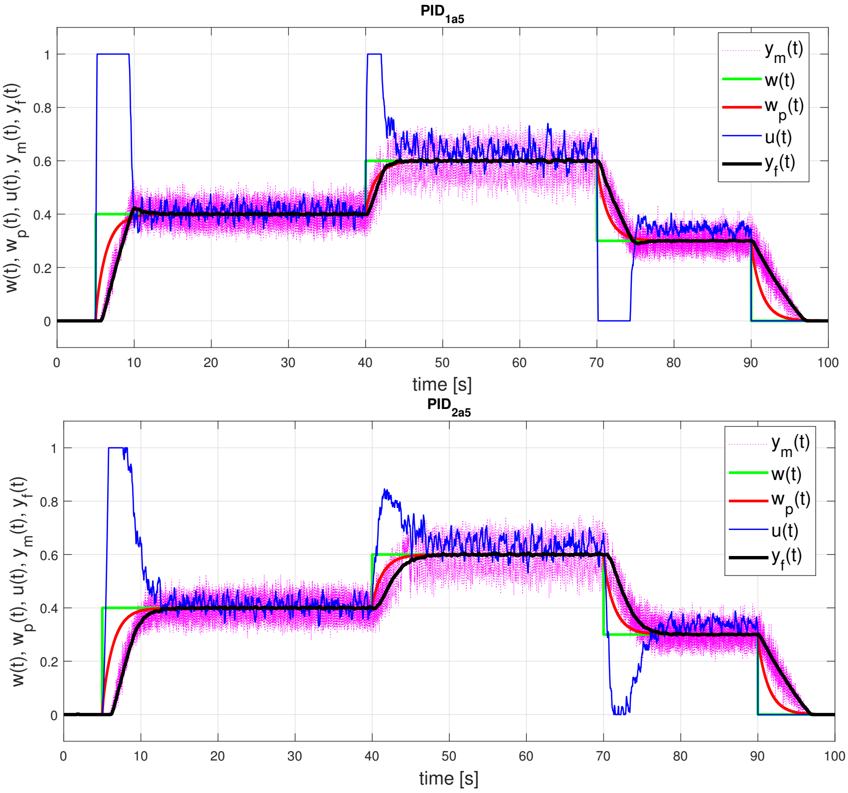

Figure 10. PID control: setpoint responses corresponding to FOTD approximations of a longer initial part of a normed step response $y(t)$ for $t_a \leq 5$ with both possible sets of the controller parameters (47).

6. Discussion

The performed experiments can be summarized in the groups of conclusions.

6.1. Step-Response-Based Plant Modeling

In Remark 4 we explained the motivation to look for approximations with a maximum value of T_d. We implemented the reasoning itself for the IPDT model, because for the FOTD model the corresponding relationships are too complex. In addition, the rate of transients depends on another parameter (a). However, the experimental results show that a different criterion will have to be sought to select the optimal FOTD model.

By analyzing the approximations at different t_a (ta = 1: 0.1: 20) and choosing the maximum value of T_d, the optimal approximation IPDT (42) was found for $t_a = 0.8$ and the optimal FOTD5 model (43) was obtained at $t_a = 4.6$. Extending the approximation with the criterion formulated in this way no longer brings some improved results.

However, what do the FOTD approximations evaluated for higher values of t_a bring? For example, the parameters of the FOTD20 model, obtained by least squares for $t_a = 20$ and parameter grid (41) are

$$\text{FOTD20},\ t_a = 20:\ K_s = 0.15;\ a = 0.161;\ T = 1/a = 6.2;\ T_d = 0.18 \tag{48}$$

This model then yields parameters of the PI and PID control

$$\begin{aligned}
&PI_{a20}:\\
&K_{Po} = 16.84;\ T_{io} = 0.965;\ b_o = 0.311\\
&PID_{1a20}:\\
&K_{Po} = 26.45;\ T_{io} = 0.594;\ T_{Do} = 0.051;\ b_o = 0.148;\ c_o = 0\\
&PID_{2a20}:\\
&K_{Po} = 2.266;\ T_{io} = 0.051;\ T_{Do} = 0.594;\ b_o = 0.148;\ c_o = 0
\end{aligned} \tag{49}$$

The corresponding transients with PI control and both options of PID control in (see Table 1) just confirm the previously discussed properties: PI control and the first option of PID control show slight output overshooting and an additional control pulse at the input. Monotonic output changes correspond just to the 2nd set of controller parameters (49). However, when decreasing the steps in looking for the optimal least squares responses in the parameters K_s and T (41) to one half and repeating the optimal model search with $t_a = 20$ and $t_a = 30$, we may see that with respect to FOTD20, the model FOTD20X changed significantly, whereas (considering the same number of decimal places) FOTD30X is nearly identical with FOTD20

$$\begin{aligned}
&\text{FOTD20X},\ t_a = 20:\ K_s = 0.145;\ a = 0.155;\ T = 1/a = 6.45;\ T_d = 0.1\\
&\text{FOTD30X},\ t_a = 30:\ K_s = 0.150;\ a = 0.161;\ T = 1/a = 6.20;\ T_d = 0.18
\end{aligned} \tag{50}$$

That is, the parameters a (or $T = 1/a$) and T_d may vary in a relatively broad range, whereas the parameter K_s remains roughly constant. All the newly calculated values of T_d are now smaller than $T_d = 0.27$, obtained at $t_a = 4.6$. Figure 11 shows that the time response of the FOTD1 and FOTD5 models resemble the actual time response of the process only in the first few seconds. Therefore, the approximation intervals $t_a \leq 1$, $t_a \leq 5$, $t_a = 20$ or $t_a = 30$ used are still short with respect to the total measurement time to steady state $t_{max} = 60$ (in Figure 4), whereas the settling time is roughly 30 s. Responses corresponding to FOTD20X and FOTD30X are graphically difficult to distinguish, but this is no longer the case for T and T_d values and even less for controller parameters.

On the one hand, this means that repeating the task for several values t_a, with matching the obtained values according to (12), loses its meaning for longer t_a. On the other hand, above results show that from a numerical point of view, the method of identifying FOTD model is poorly conditioned and should be avoided. If we extend the length of the approximation further, the more the effect of the various perturbations will become apparent, and we could perhaps reach completely unusable results and conclude that the Ziegler and Nichols method may not be practically applicable.

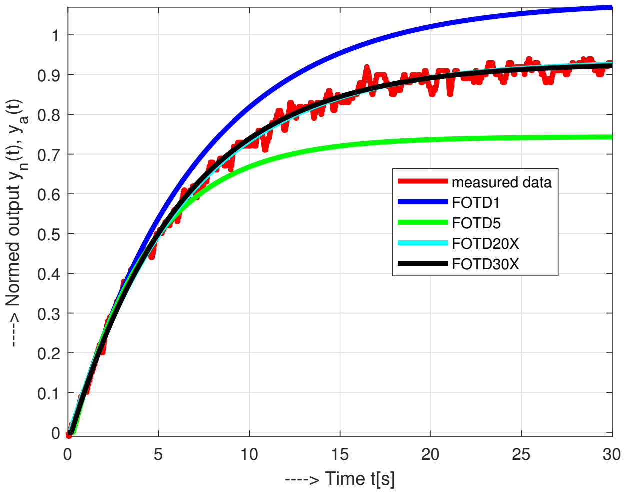

Figure 11. FOTD approximations of a normed step response $y_n(t)$ calculated for $t_a \leq 1$, $t_a \leq 5$, $t_a = 20$ and $t_a = 30$.

We may continue this analysis with evaluating parameters of the controllers corresponding to the obtained models FOTD20X and FOTD30X. For the PI controller we obtain parameters

$$\begin{aligned} PI_{a20X}: \; K_{Po} &= 31.56; \; T_{io} = 0.557; \; b_o = 0.168 \\ PI_{a30X}: \; K_{Po} &= 16.84; \; T_{io} = 0.966; \; b_o = 0.300 \end{aligned} \tag{51}$$

If we mentioned above the high numerical sensitivity of determining the parameters of FOTD models, this is even more true for determining the parameters of PI controllers. In the case of graphically difficult-to-distinguish approximations of measured responses, we obtain almost double the differences in terms of determining the optimal controller gain. In addition, there are big differences in other parameters as well, which finally lead to much more oscillatory responses in Figure 12.

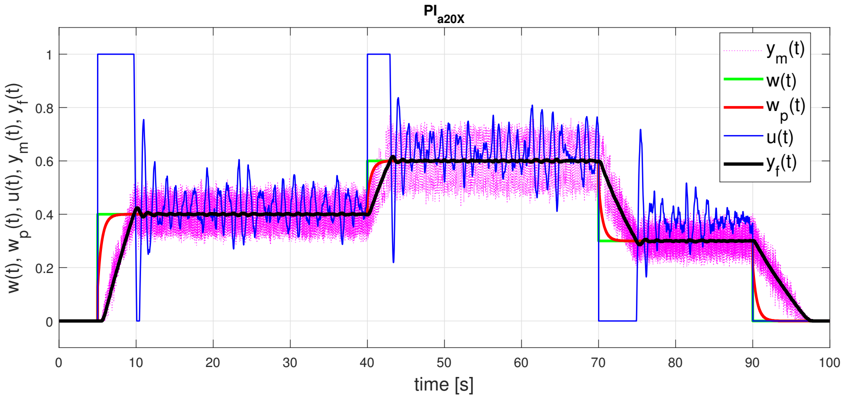

Figure 12. PI control: setpoint responses corresponding to FOTD20X approximation (50) achieved with $t_a = 20$ for a modified grid of parameters (41) with two times finer steps for T and K_s and the controller parameters (51).

Even greater differences are shown in the case of the PID controller. Here we obtain

$$\begin{aligned}
&PID_{1a20X}: \\
&K_{Po} = 35.67;\ T_{io} = 0.495;\ T_{Do} = 0.011;\ b_o = 0.078 \\
&PID_{2a20X}: \\
&K_{Po} = 0.782;\ T_{io} = 0.011;\ T_{Do} = 0.495;\ b_o = 0.078 \\
&PID_{1a30X}: \\
&K_{Po} = 26.45;\ T_{io} = 0.594;\ T_{Do} = 0.051;\ b_o = 0.140 \\
&PID_{2a30X}: \\
&K_{Po} = 2.266;\ T_{io} = 0.051;\ T_{Do} = 0.594;\ b_o = 0.140
\end{aligned} \tag{52}$$

Also these parameters lead to much more oscillatory responses in Figure 13.

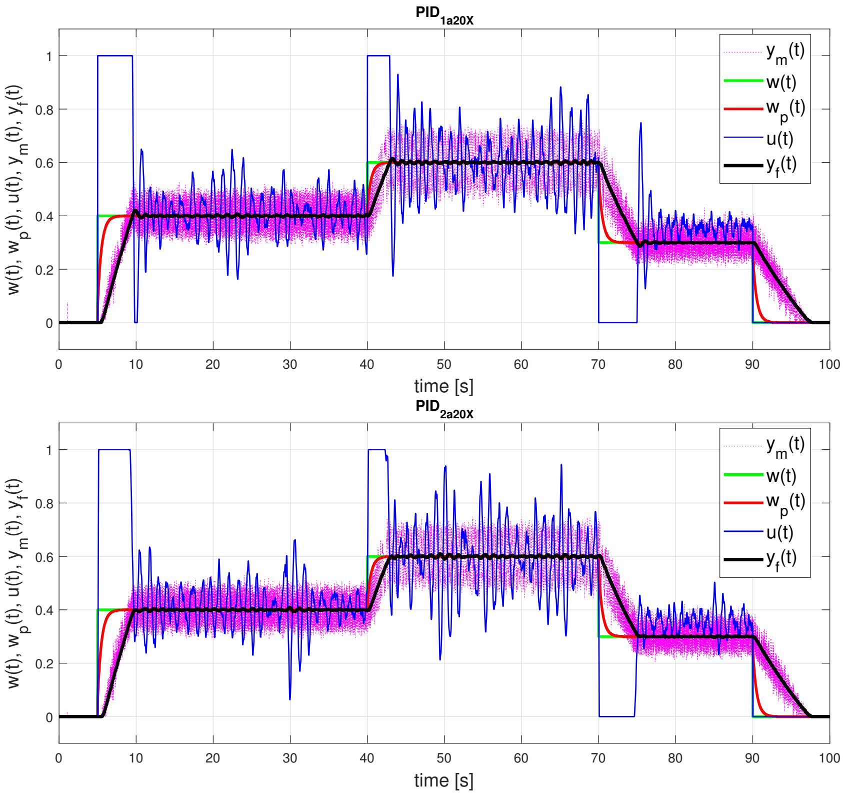

Figure 13. PID control: setpoint responses corresponding to FOTD20X approximation (50) achieved with $t_a = 20$ for a modified grid of parameters (41) with two times finer steps for T and K_s and the controller parameters (52).

So the basic question here is how do we recognize the most advantageous FOTD model? Surprisingly, the settings of both PI and PID controllers resulting from the FOTD20 and FOTD30X models are very close to the parameters calculated using the IPDT model. With the field of results of experimental verification, this points to the already mentioned principle (see Remark 3) that when determining the parameters of the FOTD model, we should consider their compliance with the parameters from the IPDT model as a criterion of their aptness, i.e., regarding the accuracy of determining the fastest mode, the optimal FOTD model has to be specified in such a way that its parameters K_s and T_d are similar for both $S_a(s)$ (6) and $S_0(s)$ (8) and thus yield also similar controller parameter values. Of course, when evaluating the suitability of the PID parameters (52) resulting from the FOTD20X model, another question arises as to what extent the selected sampling period $T_s = 0.01$ s is sufficient for their implementation.

6.2. Two Options in PID Parametrization

Additionally, the paper comes to the conclusions that the use of a more complex PID controller can give faster transients with smaller output deviations from ideal shapes, even in the case of heavily noisy processes, than can be obtained with simpler PI controllers. This example is important regarding numerous applications in mechatronics, where great emphasis is placed on the fastest possible monotonic step responses without overshooting. When achieving such transients, control action constraints usually also apply. From this point of view, the use of anti-windup I-action accomplished in the form of feedback from the constrained output of the controller (as in Figure 2) proves to be interesting. It is realized with a delayed time constant T_i. However, when using a PI controller, output overshoot almost always appears. This can simply be avoided using a series PID controller but only with one of the two possible controller parameterizations. This fact was first pointed out in [44]. The realized experiments fully confirmed expectations from [44] and so they are definitely not a result of random parasitic effects and uncertainties arising under real-time control. Although it may seem surprising and counter-intuitive that a circuit that under linear control, gives the same transients for both possible sets of controller parameters, gives different properties in limiting the control action. It should be noted, however, that to examine the effect of a nonlinear saturation block (representing a special case of sector nonlinearity), the control loop must be transformed into an equivalent canonical circuit with saturation and a linear part [17,59,60,62]. In such a case, however, different controller parametrization corresponds to different transfer functions of the linear part, which may explain the different behavior. However, since the works in this area focus on stability, while not paying attention to the monotonicity of transients, given the limited scope of this article and its focus on experiments with the Arduino computer, we will avoid more detailed considerations here. The benefit of the new solution with PID control is even more valuable because traditional anti-windup controllers based on conditioning techniques [63] always lead to overshooting [2].

6.3. Final Recommendations

Finally, if we try to summarize the results of the carried-out experiments quantified by Table 1 into the final recommendations, regarding the emphasized advantages, they will be different:

If a nearly time-optimal transient responses (low IAE) with well-damped steady states (low $TV_1(u)$) are required, while overshooting up to 10% is not a problem, PI_{a5} appears to be the most appropriate solution. PI_{a1} and PI_0 offer roughly the same output at approximately twice the excessive input increments ($TV_1(u)$). The number of samples needed to determine the model has been reduced for PI_0, which makes it possible to shorten the identification of the system. The use of FOTD20, FOTD20X or FOTD30X models with a significantly longer identification phase and more demanding parameter evaluation does not bring any obvious advantages. Rather, it is accompanied by a problem which of the obtained models, which lead to a considerable variance of the controller parameters, is actually preferred. If it is necessary to reduce output overshooting and excessive output increments as much as possible, PID controllers corresponding to the second option parameters prove to be the most advantageous. For the upwards steps, PID_{2a5} and PID_{20} are most preferred, while when jumping downwards, PID_{20} is the best. Again, the use of models with a longer approximation interval does not bring obvious advantages. If further smoothing of the responses is needed, it would be even more necessary to reduce the sampling period T_s.

7. Conclusions and Future Work

In the paper, some modifications to the oldest and most frequently cited method of setting the parameters of PI and PID controllers [3], resulting from the need to digitize them, have been proposed and tested. When trying to digitize the whole design, the first modification avoids the digital determination of the inflection point of the step response curve of the process. The proposed solution is not only numerically more stable, but also allows more accurate identification of the relevant process parameters. Instead of the original quarter-amplitude-damping, performance was evaluated based on deviations from ideal input and output responses formulated based on monotonicity. The optimal controller parameters are determined using the Multiple Real Dominant Poles method applied to PI and series PID controllers. The applied anti-windup protection ensures that the input and output responses of the process are close to the ideal shapes even for constrained systems. The study also showed that more complex FOTD process models offer little or no improvement over the IPDT models in the experiments provided. Moreover, the use of simple integrating process models, avoiding the identification of internal feedback parameters (and denoted therefore also as model-free control), is a cornerstone of modern control methods, such as advanced disturbance rejection control.

The third modification is the implementation of a low-pass filter on the measured process output. By reducing measurement noise, the filter improves the accuracy of process identification and significantly reduces the controller output noise during control. Therefore, the PID controller is also suitable for noisy processes. As already stated in some works [2,14], the control performance for noisy and uncertain processes can be further improved using controllers with higher-order derivatives.

Author Contributions: Writing-original draft preparation, M.H. and D.V. Experiments and simulations, S.C. and M.H. Figures M.H. and P.B. Editing, D.V., P.B. and M.H. Project administration, M.H. and P.B. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the grants APVV SK-IL-RD-18-0008 Platoon Modeling and Control for mixed autonomous and conventional vehicles: a laboratory experimental analysis, VEGA 1/0745/19 Control and modeling of mechatronic systems in emobility and within research program P2-0001, financed by the Slovenian Research Agency.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

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

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

Abbreviations

The following abbreviations are used in this manuscript:

1P	One Pulse, response with 2 monotonic segments (1 extreme point)
ADRC	Active Disturbance Rejection Control
ART	Average Residence Time
DC	Direct Current
IAE	Integral Absolute Error
IPDT	Integrator Plus Dead-Time
IOSSCH	Input-Output Steady-State Characteristic
MRDP	Multiple Real Dominant Pole
PID	Proportional-Integrative-Derivative
QRDP	Quadruple Real Dominant Pole
TV	Total Variation
TV_0	Deviation from monotonicity (0P shape)
TV_1	Deviation from 1P shape
TRDP	Triple Real Dominant Pole

References

1. Åström, K.J.; Hägglund, T. *Advanced PID Control*; ISA, Research Triangle Park: Durham, NC, USA, 2006.
2. Huba, M.; Vrančić, D.; Bisták, P. PID Control with Higher Order Derivative Degrees for IPDT Plant Models. *IEEE Access* **2021**, *9*, 2478–2495. [CrossRef]
3. Ziegler, J.G.; Nichols, N.B. Optimum settings for automatic controllers. *Trans. ASME* **1942**, *64*, 759–768. [CrossRef]
4. Cohen, G.; Coon, G. Theoretical investigation of retarded control. *Trans. ASME* **1953**, *75*, 827–834.
5. Hang, C.; Åström, K.; Ho, W. Refinements of the Ziegler Nichols tuning formula. *IEEE Proc. D* **1991**, *138*, 111–118. [CrossRef]
6. Žáková, K.; Huba, M. Theoretical Analysis of Ziegler-Nichols Conclusions. *IFAC Proc. Vol.* **1997**, *30*, 195–200. [CrossRef]
7. Huba, M.; Žáková, K. Contribution to the Theoretical Analysis of the Ziegler-Nichols Method. *J. Electr. Eng.* **2003**, *54*, 188–194.
8. Hägglund, T.; Aström, K.J. Revisiting the Ziegler-Nichols Tuning Rules for PI Control. *Asian J. Contr.* **2002**, *4*, 364–380. [CrossRef]
9. Åström, K.J.; Hägglund, T. Revisiting the Ziegler-Nichols step response method for PID control. *J. Process Control* **2004**, *14*, 635–650. [CrossRef]
10. Šekara, T.B.; Mataušek, M.R. Revisiting the Ziegler-Nichols process dynamics characterization. *J. Process Control* **2010**, *20*, 360–363. [CrossRef]
11. Brito, A.G. On the misunderstanding of the Ziegler-Nichols's formulae usage. *IEEE/CAA J. Autom. Sin.* **2019**, *6*, 142–147. [CrossRef]
12. Takahashi, Y.; Chan, C.; Auslander, D. Parametereinstellung bei linearen DDC-Algorithmen. *Regelungstechnik* **1971**, *19*, 237–244. [CrossRef]
13. Huba, M.; P.Bisták, Z.; Žáková, K. Predictive Antiwindup PI and PID-Controllers Based on I_1 and I_2 Models with Dead Time. *Theory Pract. Control Syst.* **1998**, *11*, 532–535.
14. Huba, M.; Vrančić, D. Extending the Model-Based Controller Design to Higher-Order Plant Models and Measurement Noise. *Symmetry* **2021**, *13*, 798. [CrossRef]
15. Feldbaum, A. *Optimal Control Systems*; Academic Press: New York, NY, USA, 1965.
16. Rodrigues, R.S.; Sombra, A.; Torrico, B.; Pereira, R.; de Almeida Filho, M.; Nogueira, F.G. Tuning rules for unstable dead-time processes. *Eur. J. Control* **2021**, *59*, 250–263. [CrossRef]
17. Lima, T.A. Contributions to the Control of Input-Saturated Systems: Time Delay and Allocation Function Cases. Ph.D. Thesis, Universidade Federal Do Ceará, Fortaleza, Brazil, 2021.
18. Huba, M.; Šimunek, M. Modular Approach to Teaching PID Control. *IEEE Trans. Ind. Electr.* **2007**, *54*, 3112–3121. [CrossRef]
19. Huba, M.; Bistak, P. Dynamic classes in the PID control. In Proceedings of the 1999 American Control Conference (Cat. No. 99CH36251), San Diego, CA, USA, 2–4 June 1999; Volume 6, pp. 3868–3872.
20. Huba, M.; Bisták, P.; Skachová, Z.; Žáková, K. P- and PD-Controllers for I_1 and I_2 Models with Dead Time. *Theory Pract. Control Syst.* **1998**, *11*, 514–519.
21. Skogestad, S. Simple analytic rules for model reduction and PID controller tuning. *J. Process Control* **2003**, *13*, 291–309. [CrossRef]
22. Grimholt, C.; Skogestad, S. Optimal PID control of double integrating processes. *IFAC-PapersOnLine* **2016**, *49*, 127–132. [CrossRef]
23. Vítečková, M.; Víteček, A. 2DOF PI and PID controllers tuning. *IFAC Proc. Vol.* **2010**, *9*, 343–348. [CrossRef]
24. Víteček, A.; Vítečková, M. Desired Model Method and Disturbance Observer for Integrating Plant with Time Delay. In Proceedings of the 2021 22nd International Carpathian Control Conference (ICCC), Velké Karlovice, Czech Republic, 31 May–1 June 2021; pp. 1–5. [CrossRef]
25. Alcántara, S.; Vilanova, R.; Pedret, C. PID Tuning: Analytical approach based on the weighted Sensitivity problem. *Rev. Iberoam. Automática Inf. Ind.* **2021**, Article in press. [CrossRef]
26. Huba, M.; Vrančić, D. Delay Equivalences in Tuning PID Control for the Double Integrator Plus Dead-Time. *Mathematics* **2021**, *9*, 328. [CrossRef]
27. O'Dwyer, A. *Handbook of PI and PID Controller Tuning Rules*, 3rd ed.; Imperial College Press: London, UK, 2009.
28. Huba, M. Designing Robust Controller Tuning for Dead Time Systems. *IFAC Proc. Vol.* **2010**, *43*, 218–225 [CrossRef]
29. Strejc, V. Näherungsverfahren für aperiodische Übergangscharakteristiken. *Regelungstechnik* **1959**, *7*, 124–128.
30. Huba, M. Triple real dominant pole tuning of a filtered PI controller. In Proceedings of the 2016 Cybernetics & Informatics (K&I), Levoca, Slovakia, 2–5 February 2016.
31. Huba, M.; Bisták, P. Quadruple real dominant pole tuning of a filtered PID controller. In *CONTROLO 2016*; Lecture Notes in Electrical Engineering; Garrido P., Soares F., Moreira A., Eds.; Springer: Cham, Switzerland, 2017; Volume 402.
32. Huba, M.; Bisták, P.; Huba, T. Filtered PI and PID control of an Arduino based thermal plant. *IFAC-PapersOnLine* **2016**, *49*, 336–341. [CrossRef]
33. Ťapák, P.; Huba, M. Laboratory Model of Thermal Plant. Identification and Control. *IFAC-PapersOnLine* **2016**, *49*, 28–33 [CrossRef]
34. Huba, M.; Žáková, K. Comparing Controllers Based on IPDT Plant Models. *IFAC-PapersOnLine* **2017**, *50*, 15710–15715. [CrossRef]
35. Huba, M.; Bisták, P.; Huba, T.; Vrančič, D. Comparing filtered PID and Smith Predictor Control of a Thermal Plant. In Proceedings of the 2017 19th International Conference on Electrical Drives and Power Electronics (EDPE), Dubrovnik, Croatia, 4–6 October 2017.
36. Huba, M.; Huba, T. *Modifications of Model Free Control to FOTD Plants*; Lecture Notes in Computer Science; Springer: Berlin/Heidelberg, Germany, 2018; pp. 1–8.

37. Huba, M. Extending Spectrum of Filtered Controllers for IPDT Plant Models. In Proceedings of the 2018 Cybernetics & Informatics (K&I), Lazy pod Makytou, Slovakia, 31 January–3 February 2018.
38. Huba, M. Performance Measures and the Robust and Optimal Control Design. *IFAC-PapersOnLine* **2018**, *51*, 960–965. [CrossRef]
39. Vrančcič, D.; Huba, M.; Oliveira, P.M. PID controller tuning for integrating processes. In Proceedings of the 3rd IFAC Conference on Advances in Proportional-Integral-Derivative Control, Ghent, Belgium, 9–11 May 2018; pp. 586–591.
40. Huba, M.; Vrančič, D. Comparing filtered PI, PID and PIDD2 control for the FOTD plants. *IFAC-PapersOnLine* **2018**, *51*, 954–959. [CrossRef]
41. Huba, M.; Vrančič, D.; Bisták, P. PID^m_n 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.
42. Huba, M.; Vrančič, D.; Bisták, P. PID^m_n Control for IPDT Plants. Part 2: Setpoint Response. In Proceedings of the 26th Mediterranean Conference on Control and Automation (MED), Zadar, Croatia, 19–22 June 2018.
43. Huba, M.; Vrančič, D. Introduction to the Discrete Time PID^m_n Control for the IPDT Plant. In Proceedings of the 15th IFAC Int. Conference on Programmable Devices and Embedded Systems, Ostrava, Crezh Republic, 23–25 May 2018; pp. 119–124.
44. Huba, M. Constrained filtered PID Controller for IPDT plants . In Proceedings of the 27th Mediterranean Conference on Control and Automation (MED), Akko, Israel, 1–4 July 2019.
45. 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.
46. Huba, M.; Žáková, K.; Bisták, P.; Hypiusová, M.; Ťapák, P. Seeking a unique view to control of simple models. In Proceedings of the 12th IFAC Symposium Advances in Control Education, Philadelphia, PA, USA, 7–9 July 2019.
47. Huba, M. Exploring PID tuning strategies considering noise impact in the IPDT plant control. In Proceedings of the 16th IFAC Int. Conference on Programmable Devices and Embedded Systems, High Tatras, Slovakia, 29–31 October 2019.
48. Huba, M. Model-based higher-order PID control design. *IFAC-PapersOnLine* **2020**, *53*, 4844–4849. [CrossRef]
49. Huba, M.; Kozák, S. From e-Learning to Industry 4.0. In Proceedings of the 2016 International Conference on Emerging eLearning Technologies and Applications (ICETA), Vysoké Tatry, Slovakia, 24–25 November 2016; pp. 103–108.
50. Vaidya, S.; Ambad, P.; Bhosle, S. Industry 4.0: A Glimpse. *Procedia Manuf.* **2018**, *20*, 233–238. [CrossRef]
51. Fliess, M.; Join, C. Stability margins and model-free control: A first look. In Proceedings of the 2014 European Control Conference (ECC), Strasbourg, France, 24–27 June 2014; pp. 454–459.
52. Fliess, M.; Join, C. An alternative to proportional-integral and proportional-integral-derivative regulators: Intelligent proportional-derivative regulators. *Int. J. Robust Nonlinear Control* **2021**, accepted.
53. Åström, K.J.; Hägglund, T. *PID Controllers: Theory, Design, and Tuning*, 2nd ed.; Instrument Society of America, Research Triangle Park: Durham, NC, USA, 1995.
54. Kothare, M.; Campo, P.J.; Morari, M.; Nett, C.N. A Unified Framework for the Study of Anti-windup Designs. *Automatica* **1994**, *30*, 1869–1883. [CrossRef]
55. Huba, M. Comparing 2DOF PI and Predictive Disturbance Observer Based Filtered PI Control. *J. Process Control* **2013**, *23*, 1379–1400. [CrossRef]
56. Vítečková, M.; Víteček, A. 2DOF PID controller tuning for integrating plants. In Proceedings of the 17th International Carpathian Control Conference (ICCC), High Tatras, Slovakia, 29 May–1 June 2016; pp. 793–797.
57. Viteckova, M.; Vitecek, A.; Janacova, D. Robustness and Muliple Dominant Pole Method. In Proceedings of the 2020 21th International Carpathian Control Conference (ICCC), High Tatras, Slovakia, 27–29 October 2020; pp. 1–4.
58. Huba, M. Performance measures, performance limits and optimal PI control for the IPDT plant. *J. Process Control* **2013**, *23*, 500–515. [CrossRef]
59. Föllinger, O. *Nichtlineare Regelungen*; R. Oldenbourg Verlag: München, Germany, 1993.
60. Glattfelder, A.; Schaufelberger, W. *Control Systems with Input and Output Constraints*; Springer: Berlin/Heidelberg, Germany, 2003.
61. Huba, M.; Bélai, I. Limits of a Simplified Controller Design Based on IPDT models. *ProcIMechE Part I J. Syst. Control Eng.* **2018**, *232*, 728–741. [CrossRef]
62. Khalil, H. *Nonlinear Systems*, 2nd ed.; Prentice Hall Int.: London, UK, 1996.
63. Hanus, R.; Kinnaert, M.; Henrotte, J. Conditioning technique, a general anti-windup and bumpless transfer method. *Automatica* **1987**, *23*, 729–739. [CrossRef]