Intelligent Settling-time based PID Tuning Algorithm for DC motor speed control

Oluwasegun Somefun, Kayode Akingbade and Folasade Dahunsi

November 19, 2019

Intelligent Settling-time based PID Tuning Algorithm for DC motor speed control

Oluwasegun Somefun*, Kayode Akingbade†, Folasade Dahunsi‡,
*Department of Computer Engineering, Federal University of Technology, Akure,Ondo, Nigeria, P.M.B 704
Email: oasomefun@futa.edu.ng
†Department of Electrical and Electronics Engineering, Federal University of Technology Akure,Ondo, Nigeria, P.M.B 704.
Email: kfakingbade@futa.edu.ng
‡Department of Computer Engineering, Federal University of Technology, Akure,Ondo, Nigeria, P.M.B 704
Email: fmdahunsi@futa.edu.ng

***Abstract*—One of the lowest level control tasks, upon which other high-level controls are dependent, is the speed control of a dc motor, especially in robotics and other manufacturing industries. Usually, tuning the parameters of the proportional-integral-derivative (PID) control for this task, employs the knowledge of process model parameters. These classical methods are powerful, but can the PID control algorithm achieve similar good control performance without using a parametric mathematical model information about the actual dc motor plant to be controlled? In this paper we propose an answer to this question. First, using the intuitive notion of unity loop gain, the closed loop PID control loop is analyzed. It then leads to an ideal or optimal close-loop model response that the PID control algorithm plus physical process loop dynamics will always be forced to follow. The final result is an intelligent tuning algorithm that integrates the use of the open-loop settling-time and time-delay values of the actual open-loop process behaviour using a fuzzy inference system. Simulation results illustrate the promise and effectiveness of the proposed tuning method in guaranteeing good closed-loop performance, without using the knowledge of a process model.**

***Index Terms*—PID, two degree-of-freedom, fuzzy inference, tuning, algorithm, loop gain, process models, intelligent control, model reference adaptive control, settling time, dead-time, DC motors, speed control.**

I. Introduction

The typical tuning of control algorithms is still based on design methods that make explicit use of mathematical models of open-loop processes and their state parameters [1]–[11].

PID control can be viewed as the bread and butter [12] of control. Compared to other control algorithms, [13] specially regarded the PID, which is inherently a cognitive feedback control law, as ubiquitous, a success story. In application to many real world problems, PID control has consistently offered a intuitive and satisfactory robust control performance [4], [6], [14]–[20]. Also compared to machine learning algorithms in production, [21], argues that 95% of the machine learning algorithms are special cases of PID control.

The structure of the PID, without any loss of generality imitates the data driven error learning and adaptation model of the ultimate biological control system, that is, the human body and mind [15], [22], [23]. Hence as argued in [21], it is machine learning in a sense, since its past data is the experience of error.

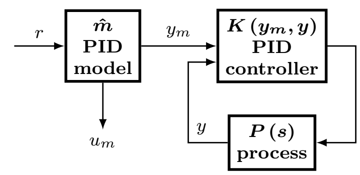

Fig. 1. PID closed-loop Model Reference – PID Controller overall block diagram.

Although, the PID has a simple general structure, it is referred to as a NP-hard problem in [24]. Tuning its three main gains for good control performance can be burdensome [7], [25], [26] even for very common servomechanism applications. In the control literature, almost all successful PID tuning algorithms embed the knowledge of an identified process or plant model. [4], [7], [8], [27]–[32]. This knowledge comes in form of a mathematical model resulting from a system identification task [33]. Process identification or plant modeling is a critical task in control design, which needs good domain understanding, as identified models are never perfect models. This imposes a limit on robustness as the tuned or set controller parameters use these imperfect model representations and so may require retuning [25], [28].

According to [34], control practitioners would prefer to avoid process identification and manual tuning of PID controllers, and so alternative realistic tuning methods are welcomed. Consequently, this motivates the need for possible alternative methodologies that can guarantee good control performance, and at the same time do not rely on parametric knowledge of an approximate mathematical model of the physical or actual open loop process. This is the area where artificial intelligence methods like fuzzy logic becomes important.

In summary, most existing tuning methods are either or both: complex, depend on the knowledge of a good plant model to work and only applicable for a specific class of systems [4], [17], [35]. Hence, according to [17], [35] there is a need for simpler methods for tuning PID controllers that can find general use in the control of a certain class of systems, such as motion control applications (high bandwidth and small input time delay).

Åström and Hägglund in [5], place the design methods for the automatic tuning of the PID control algorithm as an important research in adaptive control.

A quick highlight of the main points of this paper, are the following: it provides a unity loop gain principle approach as an intuitive basis to tuning the PID controller; it provides the derivations and analysis of the normalized model/ideal 2-DOF PID closed loop response surface; it develops a fuzzy inference knowledge system for the optimal closed-loop PID response; it designs and proposes a settling time plus delay time based performance specified tuning algorithm without using the mathematical model knowledge of the controlled process. In this way, the 2-DOF PID control algorithm [6], [29], [36]–[40] can be said to act cognitively on intuition, and also predict using the knowledge of its optimal closed-loop response.

Hopefully, this will reduce the complexity of tuning PIDs, the most applied controller [41], for dc motors which find high actuating applicability in many tasks where speed has to be controlled at the lowest level, be it in: electric vehicles, production lines, tracking systems, satellite and antenna applications, unmanned aerial vehicles, unmanned ground vehicles, robotics, computer animations, mobile phones, vehicles, chemical process control, machine tools, motor drives and many other applications [28], [42]–[45].

In the next sections, we develop a design method for realizing comparably good control performance from the proportional-integral-derivative (PID) control algorithm without using parametric mathematical model information about the plant (or process) to be controlled and then propose an automatic tuning algorithm based on this method. This paper, for presentation simplicity, is organized as follows: in section II we formulate the problem, next section III we present the analytical tuning design, then in section IV the tuning algorithm is presented, followed by section V where we evaluate the performance of the proposed tuning method on some benchmark PID processes. Finally, we conclude this paper in section VI.

II. Problem Formulation

Controllers are often implemented in a digital computer (microprocessor) as an embedded system in order to control an open-loop physical plant P(s) system as shown in Fig.2. The "(s)" notation will sometimes be removed from transfer functions for simplicity. The control output, u of the generalized continuous-form PID control algorithm function defined in (1), has a distinct signal path to the reference set-point r and the output tracked signal y as illustrated in Fig.3.

$$u\left(s\right) = \lambda_p \, K_p \left(b \, r\left(s\right) - y\left(s\right)\right) + \lambda_i \, K_i \frac{r\left(s\right) - y\left(s\right)}{s} + \lambda_d \, K_d \, s \left(c \, r\left(s\right) - y\left(s\right)\right) \tag{1a}$$

$$u = \lambda_p \, K_p \, e_p + \lambda_i \, K_i \, \frac{e_i}{s} + \lambda_d \, K_d \, s e_d \tag{1b}$$

$$u = \lambda_p \, u_p + \lambda_i \, u_i + \lambda_d \, u_d \, . \tag{1c}$$

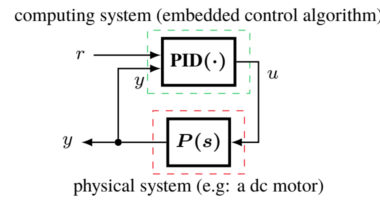

Fig. 2. Overview of an embedded PID control software loop.

where s is the complex Laplace operator, $K_p \, \epsilon \, \mathbb{R}$ is the proportional gain, $K_i \, \epsilon \, \mathbb{R}$ is the integral gain, $K_d \, \epsilon \, \mathbb{R}$ is the derivative gain, $u \, \epsilon \, \mathbb{R}$ is the control action or effort signal, $e \, \epsilon \, \mathbb{R}$ is the error signal, r is the set-point command, $y \, \epsilon \, \mathbb{R}$ is the measured plant state or output signal. Here, b and c are the proportional and derivative set-point weighting constants respectively. $\lambda_p, \lambda_i, \lambda_d$ are termed the "critic" or "reinforcement" gains for each of the three PID control terms $u_p, u_i, u_d \, \epsilon \, \mathbb{R}$ and error terms $e_p, e_i, e_d \, \epsilon \, \mathbb{R}$. We use the word "reinforcement" loosely here. The parameters $b, c, \lambda_p, \lambda_i, \lambda_d \, \epsilon \, [0, \, 1] \, \epsilon \, \mathbb{R}$. By default, $\lambda_p, \lambda_i, \lambda_d = 1$ in classical descriptions of the PID.

This kind of control structure exhibits the separation principle [3], [46] that the design problems of set-point tracking and robustness plus disturbance rejection can be achieved independently and at the same time in a control law [29], [47]. By the definition in [39], the control law's degree of freedom (DOF) is the integer-valued number of closed-loop transfer functions that are present in the controller's structure and can be adjusted independently which in this case is equal to 2 as shown in (2a). Therefore, the PID control algorithm assumed in this work is the unified parallel 2−DOF PID structure (2) which performs both set-point tracking and disturbance rejection simultaneously due to the two inherent closed loop transfer functions available in its structure using set-point limiters or weights b and c [3], [36], [38], [48]–[50]. The structure then reduces to the common 1DOF error-feedback PID structure when the set-point weights are both unity.

Simplifying (1) using (2a), a compact PID algorithm given by (2d) is obtained, where B, A, D are variables used to simplify the loop expressions in (2b) illustrated in Fig.3.

$$u = \left(K_p \, b + \frac{K_i}{s} + K_d s c\right) \, r \quad - \left(K_p + \frac{K_i}{s} + K_d s\right) \, y \tag{2a}$$

$$u = \left(K_p \, b + K_d s c\right) \; r + \frac{K_i}{s} \; e - \left(K_p + K_d s\right) \; y \tag{2b}$$

$$u = B\left(s\right) \; r\left(s\right) + A\left(s\right) \; e\left(s\right) - D\left(s\right) \; y\left(s\right) \tag{2c}$$

$$u = B \; r + A \; e - D \; y \tag{2d}$$

Tuning here, implies a mathematically sound approach to adjusting the main gains K_p, K_i and K_d of the simplified time-continuous PID structure defined in (1), without any loss of generality, in order to achieve a specified closed-loop performance for $P\left(s\right)$. In other words, finding the control gains that will push or pull a static non-linear mapping of

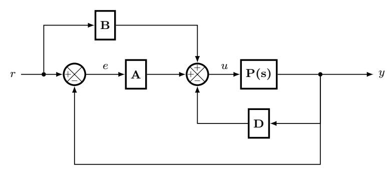

Fig. 3. Internal Block diagram of the 2DOF PID control structure.

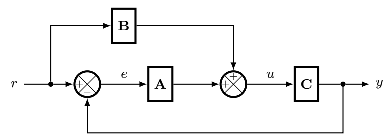

Fig. 4. Reduced block diagram of the closed loop system in Fig.3.

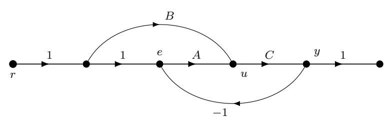

Fig. 5. Signal flow graph/map (SFG) for Fig.4.

a system to a fixed point [21]. As stated in section I, the nominal approach to designing controllers with good control performance is to set the gains of the control algorithm using identified process model parameters. The problem then is to appropriately control the actual $P(s)$ using the reduced PID expression (2b) without using an identified process model $P(s)$ as is required by conventional PID control design methods.

III. Tuning Design

The goal is to adjust the PID parameters in (1a) without using a process model $P(s)$. To achieve this, the first step is to obtain the closed loop transfer function of the overall closed loop system Fig.3.

where

$$C = \frac{P(s)}{1 + D\,P(s)} \tag{3}$$

We perform block reduction analysis (see (3) and Fig.4), then signal-flow analysis (see Fig.5) using the Mason's Gain Formula (4), where N is the number of forward paths from the input to output, P_i is the i_{th} forward path gain, Δ is the signal flow determinant, and Δ_i is the co-factor of Δ along the i_{th} forward path. Δ_i is obtained from Δ by removing the loops not touching the i_{th} forward path in Fig.5. $G(s)$, $H(s)$, and $G(s)\,H(s)$ respectively represent the overall forward path gain, the overall feedback path gain, and the loop gain or transfer function of the PID loop.

$$T(s) = \frac{y(s)}{r(s)} = \frac{G(s)}{1 + G(s)\,H(s)} = \frac{\sum_{i=1}^{N} P_i \Delta_i}{\Delta} \tag{4}$$

In this case, $N = 2$. The forward path gains are: $P_1 = AC$ and $P_2 = BC$, with a loop gain, $L_1 = -AC$. Therefore, $\Delta = 1 - L_1 = 1 + AC$, $\Delta_1 = 1$ and $\Delta_2 = 1$. Applying (5) the expression (5) for the closed loop transfer function of Fig.3 is derived.

$$T = \frac{C\,(A + B)}{1 + AC} = \frac{(A + B)\,P}{1 + (D + A)\,P} \tag{5}$$

At this point, we make the following instinctive assumptions for the closed loop characteristic equation $1 + (D + A)\,P(s)$ given by the denominator of (5):

Assumption 1: Although all physical systems are nonlinear, the process $P(s)$ is designed to work in a linear or an approximately linear operating input-output range. Then, based on the bounded-input bounded-output (BIBO) definition of stability, which states that a system is stable if every bounded input produces a bounded output, the dynamics of $P(s)$ is BIBO stable (that is, it has all its eigenvalues in the open left-half of the complex s-plane) and also affine in control.

Assumption 2: The loop gain, $(D + A)\,P(s) \gg 1$, that is, the magnitude of the loop gain will dominate the unity of its characteristic equation.

Remark 1: All negative feedback systems are based on H.S Black's 1927 idea [50]–[52] of the negative feedback amplifier. The intuitive idea according to Black to designing a successful feedback system is to make the loop gain very much larger than unity under all conditions of interest, then the closed loop gain will not be dominated by the dynamics of P(s).
Then, (5) can be approximated to (6).

$$T \to T_m = \frac{A + B}{D + A} = \frac{cs^2 + b\frac{K_p}{K_d}s + \frac{K_i}{K_d}}{s^2 + \frac{K_p}{K_d}s + \frac{K_i}{K_d}} \tag{6}$$

$$\frac{y_m}{r} = \frac{cs^2 + b2\zeta\omega_n s + \omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2} \tag{7}$$

(6) is the desired reference dynamic model response of the closed loop PID control loop, where ω_n is the natural frequency and ζ is the damping factor y_m is the desired reference output of (6).of the PID model closed loop response.

Often, the zeros in the approximate closed loop transfer function that is now the PID's model response is not desirable, so we can introduce a low pass set-point or reference input filter $F(s)$ which gives a filtered set-point $\hat{r}$ defined in (8) which transforms (6) to $\hat{T}_m(s)$ defined by (9). This is equivalent to setting both b and c to zero in (6).

$$F = \frac{\hat{r}}{r} = \frac{\frac{K_i}{K_d}}{cs^2 + b\frac{K_p}{K_d}s + \frac{K_i}{K_d}} = \frac{1}{\frac{c}{\omega_n^2}s^2 + b\frac{2\zeta}{\omega_n}s + 1} \tag{8}$$

$$\hat{T}_m = \frac{\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2} \tag{9}$$

The expected closed-loop eigenvalues (10) is obtained from the characteristic denominator equation of (7), the optimal underdamped closed loop PID model response, with ζ taking a value of 0.707, for an optimal or near optimal output response [53].

$$s_{1,2} = -\zeta\omega_n \pm j\omega_n\sqrt{1 - \zeta^2} = -\kappa\omega_d \pm j\omega_d \tag{10}$$

$$\kappa = \zeta\frac{\omega_n}{\omega_d} = \frac{\zeta}{\sqrt{1 - \zeta^2}} \tag{11}$$

where ω_d is the damped natural frequency of oscillation or imaginary part of the conjugate complex poles, κ is the exponential decay factor, and the product $\kappa\omega_d = \zeta\omega_n$ is the exponential decay frequency or real part of the poles of the closed loop PID model response.

Comparing (6) and (7), we arrive at the following nominal identities (12) and (13) for the respective gain and time-constant of the derivative and integral parts of the PID.

$$K_d = f(K_p, \omega_n, \zeta) = K_p \frac{1}{2\zeta\omega_n}, \quad T_d = \frac{1}{2\zeta\omega_n} \tag{12}$$

$$K_i = f(K_p, \omega_n, \zeta) = K_p \frac{\omega_n}{2\zeta}, \quad T_i = \frac{2\zeta}{\omega_n} \tag{13}$$

Next, it is clear from (12) and (13) that have to set the unknown closed loop's ω_n. To answer this, we analyze the output response (14) and derivative (18) of the model PID system $T_m(s)$ to a unit step input, $r(s) = \frac{1}{s}$ by applying partial fraction analysis (PFA) which gives (16) then take the inverse laplace transformation (ILT) which results to (17)

$$y_m(s) = \frac{cs^2 + b2\zeta\omega_n s + \omega_n^2}{s(s^2 + 2\zeta\omega_n s + \omega_n^2)} \tag{14}$$

$$= \frac{k_A}{s} + \frac{k_B s + k_C}{s^2 + 2\zeta\omega_n s + \omega_n^2} \tag{15}$$

where

$$k_A = 1, k_B = c - 1, k_C = (b-1)\,2\zeta\omega_n \tag{16}$$

$$y_m(t) = 1 + e^{-\zeta\omega_n t}\left[\begin{array}{l}(c-1)\cos(\omega_d t) + \\ (2b-c-1)\,\zeta\frac{\omega_n}{\omega_d}\sin(\omega_d t)\end{array}\right] \tag{17}$$

Also,

$$sy_m(s) = T_m(s) = \frac{cs^2 + b2\zeta\omega_n s + \omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2} \tag{18}$$

$$= c + \frac{k_1 s + k_2}{s^2 + 2\zeta\omega_n s + \omega_n^2} \tag{19}$$

where

$$k_1 = (b-c)\,2\zeta\omega_n, k_2 = \omega_n^2(1-c) \tag{20}$$

$$\dot{y}_m(t) = c + e^{-\zeta\omega_n t}\left[\begin{array}{l}(b-c)\,2\zeta\omega_n\cos(\omega_d t) + \\ \omega_n^2\left(\frac{(1-c)}{\omega_d} - \frac{(b-c)2\zeta^2}{\omega_d}\right) \\ \zeta\frac{\omega_n}{\omega_d}\sin(\omega_d t)\end{array}\right] \tag{21}$$

If we let the normalized time at $\omega_n = 1$, be $x = \zeta\omega_n\, t$, then (17) can be expressed compactly as (22).

$$\zeta\frac{\omega_n}{\omega_d} = \frac{\zeta}{\sqrt{1-\zeta^2}} = \kappa \tag{21a}$$

$$\omega_d t = \frac{x}{\kappa} \tag{21b}$$

$$y_m(x) = 1 + e^{-x}\left[\begin{array}{l}(c-1)\cos\left(\frac{x}{\kappa}\right) + \\ (2b-c-1)\,\kappa\sin\left(\frac{x}{\kappa}\right)\end{array}\right] \tag{22}$$

$$\dot{y}_m(x) = c + e^{-x}\left[\begin{array}{l}(b-c)\,2\zeta\cos\left(\frac{x}{\kappa}\right) + \\ \left(\frac{(1-c)}{\zeta} - (b-c)\,2\zeta\right) \\ \kappa\sin\left(\frac{x}{\kappa}\right)\end{array}\right] \tag{23}$$

In state-space controller form (24), the closed loop PID model response can be re-expressed as (25).

$$\begin{gathered}\dot{\boldsymbol{x}} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{B}\boldsymbol{r} \\ \boldsymbol{y_m} = \boldsymbol{C}\boldsymbol{x} + \boldsymbol{D}\boldsymbol{r} \\ \boldsymbol{x} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}, \dot{\boldsymbol{x}} = \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix}\end{gathered} \tag{24}$$

where,

$$\begin{gathered}\boldsymbol{A} = \begin{bmatrix} 0 & -\omega_n^2 \\ 1 & -2\zeta\omega_n \end{bmatrix} \boldsymbol{B} = \begin{bmatrix} \omega_n^2(1-c) \\ 2\zeta\omega_n(b-c) \end{bmatrix} \\ \boldsymbol{C} = \begin{bmatrix} 0 & 1 \end{bmatrix} \boldsymbol{D} = [c]\end{gathered} \tag{25}$$

Using the observable state space form (25) of the PID's closed-loop model state space, the graphical analysis of the model's output and error response becomes tractable. Further analyzing (25), and assuming a constant or slowly varying set-point command signal r, the following interesting relationships (26)–(32), can be derived.

$$y_m = x_2 + cr \therefore x_2 = y_m - cr = -e_{d,m} \tag{26}$$

$$\therefore e_{d,m} = -x_2, \dot{e}_{d,m} = -\dot{x}_2 \text{ and } \therefore \dot{y}_m = \dot{x}_2 \tag{27}$$

$$\dot{x}_1 = \omega_n^2(-x_2 + (1-c)\,r) \tag{28}$$

$$\therefore e_m = r - y_m = \frac{\dot{x}_1}{\omega_n^2} = e_{i,m} \quad \therefore \int e_{i,m} = \frac{x_1}{\omega_n^2} \tag{29}$$

$$\dot{x}_2 = x_1 + 2\zeta\omega_n(-x_2 + (b-c)\,r) \tag{30}$$

$$\therefore e_{p,m} = br - y_m = \frac{\dot{x}_2 - x_1}{2\zeta\omega_n} \tag{31}$$

leading to

$$u_m = e_{p,m} + \int e_{i,m} + \dot{e}_{d,m} \tag{32}$$

As illustrated in the normalized plots in Fig.6, the input-output surface of the PID's closed loop model reveals a non-linear mapping. Also, the response of the PID model is affected by the choice of b and c. The industrial form of the PID favours the I-PD form ($b = 0, c = 0$) equivalent to an integral plus state feedback controller, meanwhile classical form favours the PI-D form ($b = 1, c = 0$)

$$\omega_n = \frac{x_s}{\zeta t_s} \quad \text{or} \quad \omega_n = \frac{x_{pk}}{t_{pk}\sqrt{1-\zeta^2}} \tag{33}$$

Inspecting the resulting graphs Fig.6a and Fig.6b, it is easily observed that changing values of b and c, lead to a variation in the normalized peak times and settling times of the model PID. This is further observed by considering the area under the first undershoot of the PID's closed-loop model predictive output response for its normalized settling time, while for the normalized peak time, by considering the first point at which the derivative of this output response is zero.

From the relation $\omega_n = \frac{x}{\zeta t}$, the desired closed loop natural frequency ω_n at the normalized settling time $x = x_s$ with respect to the operating settling time t_s of the actual open-loop plant can be calculated. Alternatively, the desired ω_n may also be calculated using the normalized peak time $x = x_{pk}$ value, with respect to the operating open-loop peak time value t_{pk}.

Therefore, to compute ω_n using (33), the experience or knowledge of this PID's reference or optimal closed-loop

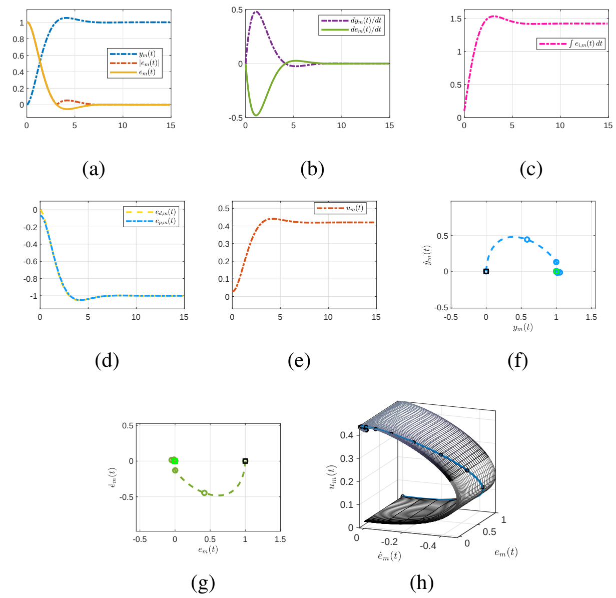

Fig. 6. Graphical state space analysis of the PID's ideal model response, $b = 0$, $c = 0$; (6a) output and error, (6b) output rate and error rate (6c) integral error, (6d) proportional and derivative errors, (6e) control output, (6f) output phase plot, (6g) error phase plot, (6h) control output surface map.

TABLE I
TYPE-I FIS DESIGN CHOICE

FIS Type	Takagi-Sugeno-Kang (TSK)	
Input(s)	b and c $\therefore p = 2$	
Antecedent Fuzzifier	Singletons	
Universe	$0 - 1$	
Output(s)	x_{pk}	x_{ts}
Consequent Fuzzifier	Singleton	Non-Singleton
Universe	$2 - 6$	$0 - 20$
T-norm	Product	Product
T-conorm	Nil	Max
MF (Parameterized)	closed n-logistic sigmoids, see (36)	
MF Parameters	Pre-specified, see Table II and III	
Number of Rules	$M = 121$ for each output	

model can be encoded as a fuzzy inference system (FIS) synthesized as a fuzzy basis function (FBF) [1], in order to automatically compute the appropriate x_s or x_{pk} with respect to the inputs b and c respectively. The design choice for this FIS is outlined in Table I, while the input-output nonlinear mapping surface of the FIS is shown in Fig.7.

Definition 1: For input $\boldsymbol{x} = [b \quad c]^T$, the Type-1 FIS that maps to $\boldsymbol{y} = [x_s \quad x_{pk}]^T$ is represented as the fuzzy basis function expansion:

$$\boldsymbol{y}(\boldsymbol{x}) = \sum_{l=1}^{M} c_0^l \phi_j^l(\boldsymbol{x}) \tag{34}$$

$$\phi_j^l(\boldsymbol{x}) = \frac{\prod_{i=1}^{p} \mu_{F_i}^l(x_i)}{\sum_{l=1}^{M} \prod_{i=1}^{p} \mu_{F_i}^l(x_i)} \tag{35}$$

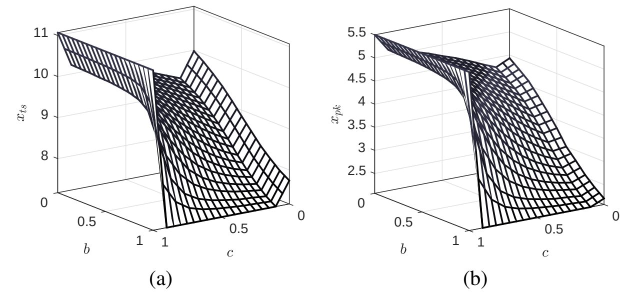

Fig. 7. Output surface of the Fuzzy Inference System (35).

$$\mu_{F_i}^l(x_i) = \begin{cases} \mu_L(x) = \text{nlsig}^-(x; \bar{c}_L, \bar{d}_L); x < \frac{\bar{c}_L + \bar{c}_R}{2} \\ \mu_R(x) = \text{nlsig}^+(x; \bar{c}_R, \bar{d}_R); x \geqslant \frac{\bar{c}_L + \bar{c}_R}{2} \end{cases} \tag{36}$$

For $nlsig^-$, the following constrains hold $x_{min}^- = \bar{c}_L - \bar{d}_L$, $x_{max}^- = \bar{c}_L$, $\bar{c}_L > x_{min}$ and $x_{min}^+ = \bar{c}_R$, $x_{max}^+ = \bar{c}_R + \bar{d}_R$, $\bar{c}_R < x_{max}$ for $nlsig^+$. Also, $y_{max} = 1$, $y_{min} = 0$.

Definition 2: The $n-$logistic sigmoid function, where $\delta \in \mathbb{R}^{n\times1}$, $\kappa_x, \kappa_y \in \mathbb{R}^{(n+1)\times1}$ with $\lambda = 6$ as a standard default value is defined as:

$$\begin{aligned} y &= \text{nlsig}^{\pm}(x; x_{min}, x_{max}, n, \lambda) \\ &= \kappa_{y,1} + \sum_{i=1}^{n} \frac{\kappa_{y,i+1} - \kappa_{y,i}}{1 + e^{\pm\alpha(x-\delta_i)}} \end{aligned} \tag{37}$$

The following holds:

$$\begin{cases} \lim_{x\to x_{max}} \text{nlsig}^-(x) = y_{max} \\ \lim_{x\to x_{min}} \text{nlsig}^-(x) = y_{min} \\ \lim_{x\to x_{min}} \text{nlsig}^+(x) = y_{max} \\ \lim_{x\to x_{max}} \text{nlsig}^+(x) = y_{min} \end{cases} \tag{38}$$

$$\Delta_x = \frac{x_{\max} - x_{\min}}{n}, \quad \Delta_y = \frac{y_{\max} - y_{\min}}{n} \tag{39}$$

$$\alpha = \lambda \frac{2}{\kappa_{x,i+1} - \kappa_{x,i}} = \lambda \frac{2}{\kappa_{x,2} - \kappa_{x,1}} \tag{40}$$

$$\delta_i = \frac{\kappa_{x,i+1} + \kappa_{x,i}}{2}, \quad i = 1,\ldots,n \tag{41}$$

$$\kappa_{x,i+1} = \kappa_{x,i} + \Delta_x, \quad \kappa_{y,i+1} = \kappa_{y,i} + \Delta_y \tag{42}$$

where $\kappa_x = [\kappa_{x,i}, \ldots, \kappa_{x,i+1}]$, $\kappa_{x,1} = x_{\min}$, $\kappa_{x,n+1} = x_{\max}$ and $\kappa_y = [\kappa_{y,i}, \ldots, \kappa_{y,i+1}]$, $\kappa_{y,1} = y_{\min}$, $\kappa_{y,n+1} = y_{\max}$.
The remaining parameter to set is K_p. For practical applications, the factors of power limitations (noise and saturation) of the hardware actuator parsing the controller's output to effect a process output, limits how high K_p can be set. A practical inquiry then, is how do we determine a stabilizing value for K_p that will effectively regulate $P(s)$ by not blowing up the actuator without resulting to the use of a process model? An intuitive answer to this, is to tune K_p as a nonlinear adaptive function of error, e and the control action u defined with the aid of extensive simulations, using the n-logistic sigmoid function in Definition (2), with, $y_{max} = x_{max}$, and $y_{min} = x_{min}$.

$$\begin{aligned} K_p &= f(e(t), u(t), x_s, x_{pk}) \\ &= nlsig^-(k_0; 0, k_{lim}, n_p, \lambda_k) \end{aligned} \tag{43}$$

TABLE II
NORMALIZED PEAK TIME: MODEL PID'S FUZZY MEMBERSHIP FUNCTION CENTERS, $\bar{c}$ [1]

c	b										
	0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1
0	4.43	4.27	4.1	3.86	3.6	3.32	3	2.78	2.55	2.36	2.2
0.1	4.29	4.15	3.99	3.79	3.54	3.26	2.96	2.68	2.44	2.23	2.07
0.2	4.41	4.3	4.15	3.97	3.73	3.44	3.11	2.78	2.49	2.26	2.07
0.3	4.54	4.44	4.32	4.15	3.94	3.65	3.3	2.92	2.57	2.28	2.07
0.4	4.66	4.58	4.48	4.34	4.15	3.89	3.54	3.11	2.68	2.33	2.07
0.5	4.76	4.7	4.63	4.52	4.37	4.15	3.83	3.37	2.85	2.39	2.07
0.6	4.87	4.82	4.77	4.69	4.58	4.41	4.15	3.73	3.11	2.49	2.07
0.7	4.96	4.93	4.89	4.84	4.77	4.65	4.48	4.15	3.54	2.68	2.07
0.8	5.04	5.02	5.0	4.97	4.93	4.87	4.76	4.58	4.15	3.11	2.07
0.9	5.12	5.11	5.1	5.09	5.07	5.04	5	4.93	4.77	4.15	2.07
1	5.5	5.5	5.5	5.5	5.5	5.5	5.5	5.5	5.5	5.5	5.5

[1] $\bar{d} = 0.01$

TABLE III
NORMALIZED SETTLING TIME: MODEL PID'S FUZZY MEMBERSHIP FUNCTION CENTERS, $\bar{c}$ [1]

c	b										
	0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1
0	9.98	9.81	9.61	9.39	9.13	8.85	8.58	8.31	8.09	7.89	7.74
0.1	9.38	9.25	9.09	8.88	8.64	8.36	8.06	7.78	7.54	7.33	7.17
0.2	9.51	9.4	9.25	9.07	8.83	8.54	8.21	7.88	7.59	7.36	7.17
0.3	9.64	9.54	9.42	9.25	9.04	8.75	8.4	8.02	7.67	7.38	7.17
0.4	9.76	9.68	9.58	9.44	9.25	8.99	8.64	8.21	7.78	7.43	7.17
0.5	9.86	9.81	9.73	9.62	9.47	9.25	8.93	8.47	7.95	7.49	7.17
0.6	9.96	9.92	9.86	9.79	9.68	9.51	9.25	8.83	8.21	7.59	7.17
0.7	10.06	10.03	9.99	9.94	9.86	9.75	9.58	9.25	8.64	7.78	7.17
0.8	10.14	10.12	10.1	10.07	10.03	9.96	9.86	9.66	9.25	8.21	7.17
0.9	10.22	10.21	10.2	10.18	10.17	10.14	10.1	10.03	9.86	9.25	7.17
1	11.07	11.07	11.07	11.07	11.07	11.07	11.07	11.07	11.07	11.07	11.07

[1] $\bar{d} = 2.22$ if $b = 0$ or $b = 1$, otherwise $\bar{d} = 2.08$

where:

$$\begin{aligned} k_0 = nlsig^{-}\left(e\left(t\right); -k_{lim}, k_{lim}, n_e, \lambda_k\right) + \\ nlsig^{-}\left(u\left(t\right); -k_{lim}, k_{lim}, n_u, \lambda_k\right). \end{aligned} \tag{44}$$

$$n_p = 16, \quad n_e = n_u = 1, \quad \lambda_k = 0.1 \tag{45}$$

$$k_{lim} = k_g \frac{2^{1.2} + 2^{1.33}}{2} \quad k_g = \begin{cases} \frac{x_s + x_{pk}}{2}, & L = 0 \\ \frac{x_s + x_{pk}}{3}, & L \neq 0. \end{cases} \tag{46}$$

With respect to setting K_p, we have used k_{lim} and shaped the n-logistic sigmoid limiter, in such a way that it constrains the set of the control gains. Therefore, provided the Assumptions 1–2 hold true, the output step response of the PID controlled loop system should then converge towards the response of the PID's model response as the term $(D + A)$ is sufficiently increased, where the increase of $(D + A)$ depends on the constrained global control gain parameter K_p defined with (43). K_p is global since it affects the value of the other control gains.

IV. TUNING ALGORITHM

The fact that the output response of a BIBO stable physical system does not instantaneously settle to a steady state value, due to the inherent input energy dissipation of physical systems [23], leads also to the intuitive choice of the settling time and delay time of the open-loop physical process to be controlled in setting the desired closed-loop natural frequency ω_n.

The results of the tuning design in the previous section, shaped with the aid of extensive simulations in MATLAB can be implemented as an algorithm on a computer processor by the following procedures:

1) Require: $L, \bar{t}_s, \zeta, b, c, e\left(t\right), u\left(t\right)$
2)
$$t_s = \begin{cases} 1 + L, & L > \bar{t}_s \\ \left|\bar{t}_s - \left(\frac{L}{\bar{t}_s}\right)\right| + L, & L \leq \bar{t}_s \end{cases} \tag{47}$$
3)
$$[x_s,\ x_{pk}] = f\left(b,\ c\right) \tag{48}$$
$$\omega_n = \frac{x_s}{\zeta t_s} \tag{49}$$
4)
$$K_p = f\left(e\left(t\right), u\left(t\right), x_s, x_{pk}\right), \quad \lambda_p = 1 \tag{50}$$

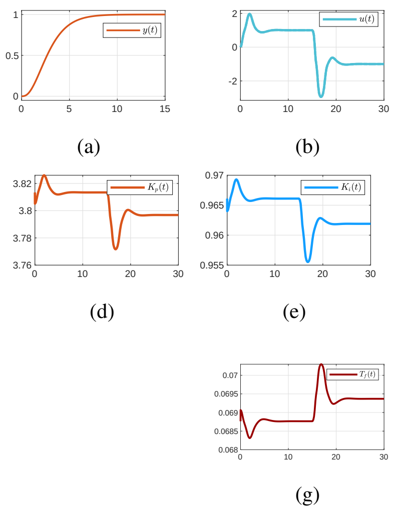

Fig. 8. Unit step response plot of $P_1\left(s\right)=\frac{1}{(s+1)^3}$; (8a) open-loop output response, (8b) control output, (8c) controlled output response, (8d-8g) tuned PID gains K_p, K_i, K_d, T_f respectively..

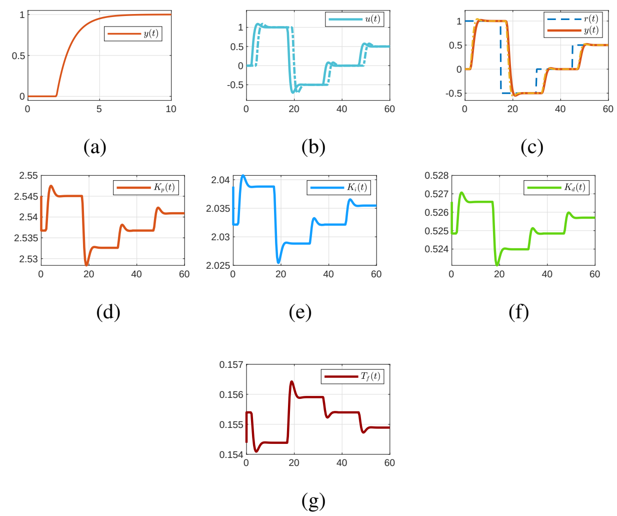

Fig. 9. Unit step response plot of $P_2\left(s\right)=\frac{1}{(s+1)}e^{-2s}$; (9a) open-loop output response, (9b) control output, (9c) controlled output response, (9d-9g) tuned PID gains K_p, K_i, K_d, T_f respectively.

5)

$$K_i = f\left(K_p, \omega_n, \zeta\right) \tag{51}$$

$$\lambda_i = \begin{cases} \lambda_i^1, & L = 0 \\ \lambda_i^2, & L > 0 \end{cases}; \quad \text{where, } \lambda_i^2 \geq \lambda_i^1 > 0 \tag{52}$$

6)

$$\bar{K}_d = f\left(K_p, \omega_n, \zeta\right) \tag{53}$$

$$K_d = \begin{cases} \frac{\bar{K}_d}{t_s(L+t_s)}, & L > t_s \\ \bar{K}_d, & L \leq t_s \end{cases} \tag{54}$$

$$\lambda_d = \begin{cases} n_d\, x_s, & L > 0 \\ 0.2, & L = 0 \end{cases} \text{ by default}, n_d = 1 \tag{55}$$

7)

$$T_f = f\left(K_p\right) = \frac{1}{K_p^2} \tag{56}$$

8) End.

V. Simulation Results and Discussion

The proposed algorithm in section IV is validated by some simulation results. In the following, simulation results will be presented and discussed.

To compare this tuning algorithm with other methods, we refer to the standard benchmark processes provided in [54], widely used by controller manufacturers for a long time as test cases for their controllers and in many research evaluations of PID control methods in literature. In this work, we restrict the use of identified process models as representations of the true process for simulation purposes and not for the control design.

The first test-case is a process described by $P_1\left(s\right)=\frac{1}{(s+1)^3}$. Fig. 8 shows the time domain output and control effort responses to a unit step regulation, and the variation of the PID control gains. It is deduced from the results in Fig. 8 that this method gives better results compared to the results of the settling-time based tuning method in [35] and similar results compared to the frequency response based fractional-order PID tuning method in [55].

Next, Fig. 9 shows the time domain output and control effort responses to a unit step regulation, and the variation of the PID control gains for a dead-time dominated normalized first-order process $P_2\left(s\right)=\frac{1}{(s+1)}e^{-2s}$. It is deduced from the results in Fig. 9 that this method gives far better results compared to the results of the settling-time based tuning method in [35].

To illustrate the potential of this method with respect to the angular speed control of a dc motor. We test the algorithm on two different dc motor models. Fig. 10 shows the control of a dc motor $P_3(s)$ reported in [56] with physical model specifications, shown in Table IV but with addition of a transport lag of 4 seconds, and a $\pm 20\%$ noise variation of the model's state, hence it can be concluded that this method is robust to parametric uncertainties and transport lag. It can be inferred from the plots in Fig. 10 that this method is very capable of handling dead-time systems and noisy models. The control of the second dc motor model $P_4(s)$, a Mitsumi 448 PPR identified in [27] is illustrated in Fig. 11. The use of this proposed algorithm shows better and comparable performance in the control of the angular speed of the dc motor compared to the fuzzy PID control and classical PID control approaches employed in [27].

For all of the test cases provided in Fig. 8–11, the output response follows the closed-loop model output response of the PID loop designed using the methods employed in section III with respect to the correct estimations of the final settling time and time-delay of the open loop process.

PID control itself can be viewed as a cognitive imitation of nature's approach to control, so its algorithm should be

TABLE IV
SPECIFICATIONS OF A DC MOTOR FOR ANGULAR SPEED

Name	Description	Value
J_m	moment of inertia	$0.01\,\mathrm{Kg.m^2}$
B_m	viscous Friction constant	$0.1\,\mathrm{N.m.s}$
K_t	motor torque constant	$0.01\,\mathrm{V.s/rad}$
K_b	electromotive force constant	$0.01\,\mathrm{N.m/A}$
R_a	armature resistance	$1\,\Omega$
L_a	armature inductance	$0.5\,\mathrm{H}$

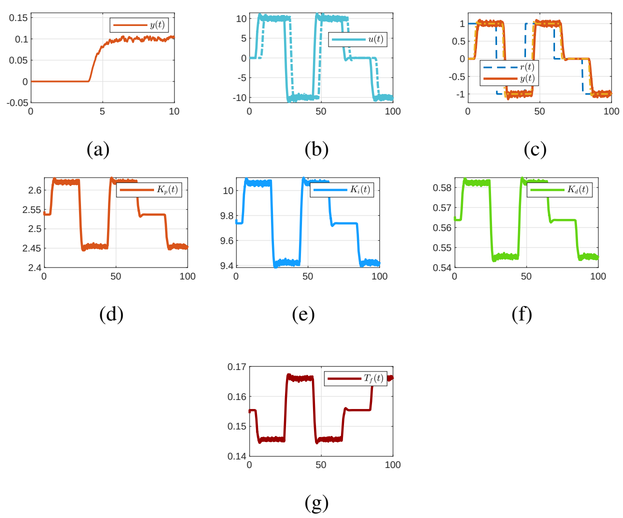

Fig. 10. Unit step response plot; (10a) open-loop output response, (10b) control output, (10c) controlled output response, (10d-10g) tuned PID gains K_p, K_i, K_d, T_f respectively for $P_3\left(s\right) = \frac{K_t}{(J_m s+B_m)(L_a s+R_a)+K_t K_b} e^{-4s}$.

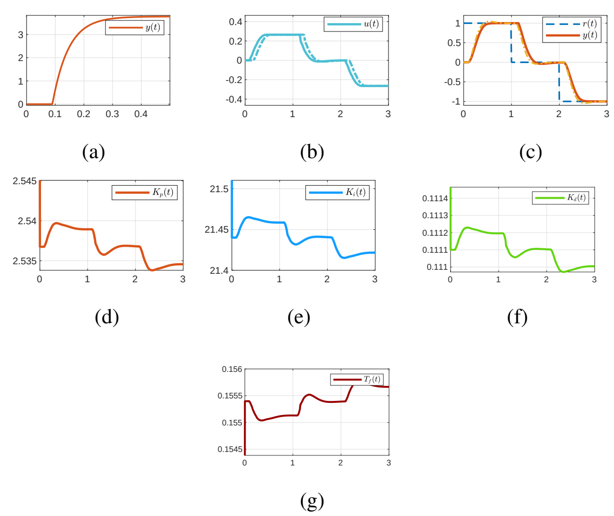

Fig. 11. Unit step response plot of $P_4\left(s\right) = \frac{3.776}{(0.56s+1)} e^{-0.09s}$; (11a) open-loop output response, (11b) control output, (11c) controlled output response, (11d-11g) tuned PID gains K_p, K_i, K_d, T_f respectively.

intelligently based on the fore-knowledge of its ideal or model closed-loop performance. The proposed method in this paper, as outlined by the resulting algorithm in section IV is a modest start-point in an attempt to designing PID control structures and tuners that tightly integrate with artificial intelligent methods and do not rely on the knowledge of identified model approximations of a physical process as noted in [2]. The PID is often described as a linear controller in literature. This statement is neither completely true nor false. The interpretation of the input-output surface of the PID's closed-loop model in Fig. 6h is that the PID loop's control surface is a convex map that can become approximately linear, and so therefore the PID control scheme may not be as simple or linear as is the common belief. There are still many insights that can be gained from the PID's closed-loop model surface especially for designing possible reinforcement learning and adaptive control algorithms.

Two key performance metrics for a control system are overshoot and settling time [35], [57]–[59]. According to [14], engineering practice is persistently demanding for PID control design methods that simultaneously guarantee these two metrics. Using the understanding of the behaviour of the open loop process at its operating region, we easily embed these performance specifications in the optimal closed loop PID model response.

In itself process identification is a must, as it helps control designers understand the limits and input-output behaviour of the process, but by restricting the use of process models to testing the performance of the tuned control algorithm, we may be able to design truly intelligent control. For intelligent PID control, tuning of the control gains, should not rely on the knowledge of the mathematical model parameters of an open-loop process, more so, especially for many common processes that are not difficult to control with respect to stability and dead time, for example: dc servomotors, then regulating such processes around their operating regions becomes less complex and more automatic, as the time spent on looking for good and near optimal control gains is reduced significantly compared to existing and traditional auto tuning methods based on model identification [33]. Therefore, we conclude that this tuning method has potential for use in place of the popular Ziegler-Nichols tuning rule, as a first try for tuning PIDs.

VI. CONCLUSION

In this paper, we have proposed an automatic PID tuning algorithm, designed with its foundation being the intuitive assumptions on the unity loop gain principle. Just like in fuzzy-logic controls, this control tuning design is not based on the traditional mathematical model-based tuning control design. It leads to an automatic algorithm that industry engineers and non-expert users can employ to find good PID control. To validate the proposed tuning method, the method was verified on benchmark systems and compared with those in literature. From the results, we surmise that this tuning algorithm has potential for wide applicability to many PID controllable plants even when dead-time dominated. A promising practical industrial application of this developed algorithm is in the robust speed control of dc motors.

Acknowledgment

This research was not funded at anytime. However, we acknowledge the shoulders of the many cited researchers whose easily accessible works helped solidify this research.

References

[1] J. M. Mendel, *Uncertain Rule-Based Fuzzy Systems: Introduction and New Directions, 2nd Edition*, 2nd ed. Springer International Publishing, 2017.

[2] L. Li, N. Zheng, and F. Wang, "On the Crossroad of Artificial Intelligence: A Revisit to Alan Turing and Norbert Wiener," *IEEE Transactions on Cybernetics*, vol. 49, no. 10, pp. 3618–3626, Oct. 2019.

[3] K. J. Åström and T. Hägglund, "Design methods: PID Control," *The Control Handbook: Control System Fundamentals, Second Edition*, pp. 9–70–9–90, 2017.

[4] S. Li and J. Wang, "Research on Engineering Tuning Methods of PID Controller Parameters and Its Application," in *Intelligent Computing Methodologies*, ser. Lecture Notes in Computer Science, D.-S. Huang, K. Han, and A. Hussain, Eds. Springer International Publishing, 2016, pp. 563–570.

[5] W. S. Levine, *The Control Systems Handbook: Control System Advanced Methods, Second Edition (Electrical Engineering Handbook)*, 2nd ed., ser. Electrical Engineering Handbooks. CRC Press, 2011.

[6] S. W. Sung, J. Lee, and L. In-Beum Lee, *Process Identification and PID Control*. John Wiley & Sons, Ltd, 2009.

[7] N. J. Killingsworth and M. Krstic, "PID tuning using extremum seeking: Online, model-free performance optimization," *IEEE Control Systems Magazine*, vol. 26, no. 1, pp. 70–79, Feb. 2006.

[8] A. Visioli, *Practical PID Control*. Springer Science & Business Media, Nov. 2006.

[9] P. J. Antsaklis, "Intelligent learning control," *IEEE Control Systems Magazine*, vol. 15, no. 3, pp. 5–7, 1995.

[10] P. J. Antsaklis, K. M. Passino, and S. J. Wang, "An introduction to autonomous control systems," *IEEE Control Systems Magazine*, vol. 11, no. 4, pp. 5–13, Jun. 1991.

[11] ——, "Towards intelligent autonomous control systems: Architecture and fundamental issues," *Journal of Intelligent and Robotic Systems*, vol. 1, no. 4, pp. 315–342, Dec. 1989.

[12] G. J. Silva, A. Datta, and S. P. Bhattacharyya, *PID Controllers for Time-Delay Systems*, ser. Control Engineering. Birkhäuser Basel, 2005.

[13] T. Samad, "A Survey on Industry Impact and Challenges Thereof [Technical Activities]," *IEEE Control Systems Magazine*, vol. 37, no. 1, pp. 17–18, Feb. 2017.

[14] Š. Bucz and A. Kozáková, "Advanced Methods of PID Controller Tuning for Specified Performance," *PID Control for Industrial Processes*, Sep. 2018.

[15] M. Chevalier, M. Gomez-Schiavon, A. Ng, and H. El-Samad, "Design and analysis of a Proportional-Integral-Derivative controller with biological molecules," *bioRxiv*, Apr. 2018.

[16] Y. Peretz, "A Randomized Algorithm for Optimal PID Controllers," *Algorithms*, vol. 11, no. 6, p. 81, Jun. 2018.

[17] A. A. Dastjerdi, N. Saikumar, and S. H. HosseinNia, "Tuning guidelines for fractional order PID controllers: Rules of thumb," *Mechatronics*, vol. 56, pp. 26–36, Dec. 2018.

[18] K. Soltesz and A. Cervin, "When is PID a good choice?" *IFAC-PapersOnLine*, vol. 51, no. 4, pp. 250–255, Jan. 2018.

[19] W. Yu, "Chapter 8 - PID Admittance Control in Task Space," in *PID Control with Intelligent Compensation for Exoskeleton Robots*, W. Yu, Ed. Academic Press, Jan. 2018, pp. 139–158.

[20] K. H. Ang, G. Chong, and Y. Li, "PID control system analysis, design, and technology," *IEEE Transactions on Control Systems Technology*, vol. 13, no. 4, pp. 559–576, Jul. 2005.

[21] B. Recht, "A Tour of Reinforcement Learning: The View from Continuous Control," *arXiv:1806.09460 [cs, math, stat]*, Jun. 2018.

[22] A. O'Dwyer, "PID control: The early years," Cork Institute of Technology, Cork, Mar. 2005.

[23] A. Tewari, *Modern Control Design: With MATLAB and SIMULINK*. Wiley, Apr. 2002.

[24] L. Koszaka, R. Rudek, and I. Pozniak-Koszalka, "An Idea of Using Reinforcement Learning in Adaptive Control Systems," in *International Conference on Networking, International Conference on Systems and International Conference on Mobile Communications and Learning Technologies (ICNICONSMCL'06)*, Apr. 2006, pp. 190–190.

[25] T. Roux-Oliveira, L. R. Costa, A. V. Pino, P. Paz, T. Roux-Oliveira, L. R. Costa, A. V. Pino, and P. Paz, "Extremum Seeking-based Adaptive PID Control applied to Neuromuscular Electrical Stimulation," *Anais da Academia Brasileira de Ciências*, vol. 91, Jan. 2019.

[26] C. Grimholt and S. Skogestad, "Optimal PI and PID control of first-order plus delay processes and evaluation of the original and improved SIMC rules," *Journal of Process Control*, vol. 70, pp. 36–46, 2018.

[27] Y. Bai and Z. S. Roth, *Classical and Modern Controls with Microcontrollers: Design, Implementation and Applications*, ser. Advances in Industrial Control. Springer International Publishing, 2019.

[28] K. M. Lynch, N. Marchuk, and M. L. Elwin, "PID Feedback Control," in *Embedded Computing and Mechatronics with the PIC32*. Elsevier, 2016, pp. 375–385.

[29] A. Forrai, *Embedded Control System Design: A Model Based Approach*. Heidelberg ; New York: Springer, 2013, oCLC: ocn811007710.

[30] M. Li, J. Wang, and D. Li, "Performance robustness comparison of two PID tuning methods," in *Proceedings of the 29th Chinese Control Conference*, Jul. 2010, pp. 3601–3605.

[31] A. O'Dwyer, *Handbook of PI and PID Controller Tuning Rules*. Imperial College Press, 2009.

[32] V. Aggarwal and U. O'Reilly, "A Self-Tuning Analog Proportional-Integral-Derivative (PID) Controller," in *First NASA/ESA Conference on Adaptive Hardware and Systems (AHS'06)*, Jun. 2006, pp. 12–19.

[33] J. Böling, *Identification for Control with Application to Ill-Conditioned Systems*. AAbo Akademi University Turku, Finland, 2001.

[34] K. M. Moudgalya, *Digital Control*. Wiley Online Library, 2007.

[35] J. Jantzen and C. Jakobsen, "Turning PID controller tuning into a simple consideration of settling time," in *2016 European Control Conference (ECC)*, Jun. 2016, pp. 370–375.

[36] K. Sladká, J. Czebe, and R. Wagnerová, "The 2DOF controller implementation in control system with single-chip computer," in *2018 19th International Carpathian Control Conference (ICCC)*, May 2018, pp. 228–234.

[37] X. Wang, X. Yan, D. Li, and L. Sun, "An Approach for Setting Parameters for Two-Degree-of-Freedom PID Controllers," *Algorithms*, vol. 11, no. 4, p. 48, Apr. 2018.

[38] M. Viteckova and A. Vitecek, "2DOF Controller Tuning," *International Journal of Engineering Research in Africa*, vol. 18, pp. 57–64, 2015.

[39] M. Araki and H. Taguchi, "Two-degree-of-freedom PID controllers," *International Journal of Control, Automation, and Systems*, vol. 1, no. 4, pp. 401–411, 2003.

[40] K. Astrom and T. Hagglund, *PID Controllers*, 2nd ed. Instrument Society of America, 1995.

[41] V. M. H. Guzmán and R. S. Ortigoza, *Automatic Control with Experiments*, ser. Advanced Textbooks in Control and Signal Processing. Springer International Publishing, 2019.

[42] W. Yu, *PID Control with Intelligent Compensation for Exoskeleton Robots*. Elsevier, 2018.

[43] B. Siciliano and O. Khatib, Eds., *Springer Handbook of Robotics*, 2nd ed., ser. Springer Handbooks. Springer International Publishing, 2016.

[44] "Control System Design Guide," in *Control System Design Guide (Fourth Edition)*, G. Ellis, Ed. Boston: Butterworth-Heinemann, Jan. 2012, p. iii.

[45] W. Yu, *Recent Advances in Intelligent Control Systems*. Springer, 2009.

[46] W. Wang, "The New Design Strategy on PID Controllers," in *PID Controller Design Approaches - Theory, Tuning and Application to Frontier Areas*, M. Vagia, Ed. InTech, Mar. 2012.

[47] L. Keviczky and C. Bányász, *Two-Degree-of-Freedom Control Systems*. Elsevier, 2015.

[48] B. G. Abdelaty, A. H. Ahmed, and A. N. Ouda, "Fixed Set Point Weighting 2DOF PID Controller for Control Processes," *Engineering Mathematics*, vol. 2, no. 1, p. 21, Jul. 2018.

[49] M. Vagia, "PID Controller Design Approaches-Theory, Tuning and Application to Frontier Areas," 2012.

[50] K. J. Åström, T. Hägglund, and K. J. Astrom, "Advanced PID control," 2006.

[51] J. Mbihi, *Analog Automation and Digital Feedback Control Techniques*, 1st ed., ser. Systems and Industrial Engineering. Wiley-ISTE, 2018.

[52] M. Thompson, *Intuitive Analog Circuit Design, Second Edition*, 2nd ed. Newnes, 2013.
[53] C. Yi, "Correcting the minimum ITAE standard forms of zero-displaceemnt-error systems," *Journal of Zhejiang University (Natural Science)*, vol. 23, no. 4, pp. 550–559, 1989.
[54] K. J. Åström and T. Hägglund, "Benchmark Systems for PID Control," *IFAC Proceedings Volumes*, vol. 33, no. 4, pp. 165–166, Apr. 2000.
[55] A. Brankovic, J. Guanetti, D. Selmanaj, and A. Leva, "Applying simple PID tuning rules with extended frequency response knowledge," in *2016 IEEE 55th Conference on Decision and Control (CDC)*, Dec. 2016, pp. 7573–7578.
[56] B. Messner, R. Hill, and J. Taylor, "Control Tutorials for MATLAB and Simulink (CTMS)," http://ctms.engin.umich.edu/CTMS, 2014.
[57] D. P. Atherton, "Setting the Parameters of Proportional–Integral–Derivative Controllers," *Measurement and Control*, vol. 48, no. 9, pp. 273–277, Nov. 2015.
[58] Y. Ashida, S. Wakitani, and T. Yamamoto, "Design of a Performance-Adaptive 1-Parameter Tuning PID Controller," *Proceedings of International Conference on Artificial Life and Robotics*, vol. 23, pp. 326–329, Feb. 2018.
[59] K. Ogata and Y. Yang, *Modern Control Engineering*. Prentice hall India, 2002, vol. 4.