Speed Control of DC Motors: Optimal Closed PID-Loop Model Predictive Control

Oluwasegun Somefun¹,*, Kayode Akingbade², Folasade Dahunsi¹

1Department of Computer Engineering, Federal University of Technology, Akure, Ondo, Nigeria
2Department of Electrical and Electronics Engineering, Federal University of Technology Akure, Ondo, Nigeria

Received December 8, 2019; Revised January 23, 2020; Accepted February 3, 2020

Copyright ©2020 by authors, all rights reserved. Authors agree that this article remains permanently open access under the terms of the Creative Commons Attribution License 4.0 International License

Abstract One of the lowest level control tasks, especially in robotics and other manufacturing industries, upon which other high-level controls are dependent, is the speed control of a dc motor. Usually, tuning the parameters of the proportional integral derivative (PID) controller for this task employs established conventional methods that expose the knowledge of nominal process model parameters to the control algorithm. These methods have found widespread use. Notwithstanding, a promising line of inquiry is: to search alternate possibilities of a PID being designed to automatically achieve comparable good control performance without using a formal mathematical process model approximation of the actual physical system, such as dc motor plant, in the frequency or time domain. In this paper, we propose an intelligent PID design method, “optimal closed PID-loop model predictive control”, that answers this question using the characteristic settling-time (including delay-time) property of dynamic processes. The performance of this proposed method is benchmarked with popular process-model based methods. Simulation results illustrate the promise and effectiveness of the proposed tuning method, in ensuring good closed-loop performance quality for the dc motor, without using formal process models.

Keywords PID Controller, Algorithm, Adaptive Control, Optimal Control, Intelligent Control, Robust Control, DC Motor

1 Introduction

A generalized form of the proportional integral derivative PID structure, often referred to as the two degree-of-freedom (2DOF)-PID [1–7] fully captures the PID controller in a state-feedback plus integral form. In this paper, we introduce a 2DOF-PID controller design method, called “optimal closed PID-loop model predictive control” (OCPID-LMPC), for the intelligent speed control of a dc motor. In many applications; for instance, in mobile robotics, the speed control of a dc motor is critical to the performance of the whole system. The controller of choice for this task is mostly a PID. By conventional design methods, the PID controller heavily makes direct use of the knowledge of a nominal process model. The parameters of this formal approximation of the actual process are then used as design constants in the control algorithm [5,8–14].

Related Works. Åström and Hägglund, in [12], place the development of design methods for the automatic tuning of the PID control algorithm as an active research in adaptive control. The tuning problem of searching for optimal PID parameters for a control task is NP-hard (nondeterministic polynomial-time) [15]. Hence, PID tuning for good control performance can be burdensome [13,16,17] even for very common servomechanism applications.

It is underscored in [5,18–21] that successful industry applications always require deep domain knowledge and understanding of process identification (plant or system modeling) [4,11,13,14,22–28]. Unfortunately, situations arise when these established control design methods become unreliable because of their basis on imperfect representations of the actual process dynamics. To remedy this problem, some researchers advocate for computational intensive online parameter estimations and optimizations [13,27,29].

Control practitioners [30], would prefer to avoid identification and manual tuning of PID controllers, and so alternative realistic tuning methods are welcomed. To answer this need, many model-free approaches have been proposed in literature, of note, are methods such as the unfalsified method, iterative feedback tuning, extremum-seeking method [13].

Most recent works on PID tuning in the literature such as [31–35] are devoted to employing metaheuristic optimization algorithms as new PID tuning methods. This way, different objective functions and performance constraints such as stability, bandwidth, and robustness are considered for tuning the PID. However, these methods are often carried out offline using the nominal process model, and so any serious uncertainty, as can be found in practice, in the parameters of the system can lead to instability. Again, in some cases, the optimal solution may
never be found [27]. Therefore these methods have not found scalable use in the industry [36].

The OCPID-LMPC method, in this paper, generalizes the settling-time based method in [28]. It differs from the process-model based design convention in the PID literature [7]. Instead, it is based on a critic 2DOF-PID structure, and the characteristic delayed settling-time property of dynamic processes. Using this approach, we show, in this paper, that the PID can be designed to automatically achieve good quality of control performance by using only the settling time of the actual dc motor plant dynamics.

**Contributions.** Naturally, control intelligence can be viewed as PID control. The PID control idea imitates the nature of criticism-based human cognitive actions based on the intuitive experience of error. It has been found in [37] that most machine learning algorithms in real-life applications are special cases of PID control structures.

**Critic PID.** As the synthesis of intelligence in automatic control algorithms continues to be a major research endeavour [9, 11, 37–39]; a future direction in controller design [9] are methods that integrate the wealth of concepts and paradigms in classical and modern controls with the science of intelligence to advance the synthesis of intelligent controllers, which will improve productivity of processes in the world.

Although, intelligent approaches that explore fuzzy logic, neural network, and proposed reinforcement learning architectures for PID design and tuning exist, these methods do not consider the idea of the PID as a critic-based cognitive structure [40]. In this paper, we show how the PID structure can be formulated in a form that more closely reflects the criticism in human learning by introducing critic weights that criticize the contributions of the three-term 2DOF-PID structure. This proposal is a future basis for fusing inference and learning by means of critic or reinforcement weights into the PID structure.

**Optimal Closed PID-Loop Model Predictive Control.** It is important to note that the contribution of this paper does not follow the approaches of conventional fuzzy-logic PID control techniques and direct model reference adaptive methods in the control literature. In this paper we show a nominal process model-free adaptive method for systematically optimizing adaptive methods in the control literature. In this section, the control problem is formulated by introducing critic gains for each of the three PID contributing terms to loosely reflect a reinforcement learning idea.

![Figure 1. Overview of an embedded PID control software loop.](image)

**Problem Formulation**

In this section, the control problem is formulated by introducing critic gains for each of the three PID contributing terms to loosely reflect a reinforcement learning idea.

**2.1 Critic PID Controller**

Consider a stable linear time-invariant (LTI) and single-input single-output (SISO) plant with unknown dynamic structure in controllable canonical state-space form (1), which can be compactly represented as \( P(s) \), as shown, in Figure 1.

\[
\begin{align*}
\dot{x}_p &= A_p x_p + B_p u + \Phi \\
y &= C_p x_p
\end{align*}
\]

where \( x_p \in \mathbb{R}^{n_x} \times 1 \) represents \( n_x \) states of the plant, \( u \in \mathbb{R}^1 \) is the bounded single control input to the plant, \( y \in \mathbb{R}^1 \) is the bounded single output state of the plant, \( A_p \in \mathbb{R}^{n_x \times n_x} \) is a unknown stable state matrix, \( B_p \in \mathbb{R}^{n_u \times 1} \) is the command input matrix, \( C_p \in \mathbb{R}^{1 \times n_x} \) is the output matrix and \( \Phi \in \mathbb{R}^1 \) is the unknown lumped uncertainty (noise, and disturbance) affecting the plant’s operating conditions.

Feedback control is motivated by uncertainty. Dealing with uncertainty is a practical issue in nature and control design. Nonlinear (robust and adaptive) control techniques are used to deal with this problem in the literature. In most practical cases, this uncertainty is fuzzy and its exact magnitude cannot be measured. However, the controller must be able to operate properly in the presence of a wide range of uncertainty.

Closer to how regulation occurs in nature, the PID control idea is a computational imitation of natural robust adaptive neural feedback control intelligence. Controllers are often implemented as an embedded software in order to control the open-loop plant or process \( P(s) \). To control the output state \( y \) of this plant, consider the critic 2DOF PID structure in (2).

In (2), the introduced terms \( \lambda_p, \lambda_1, \lambda_d \in \mathbb{R}^+ \) are termed the “critic” weights for each of the three PID contributing terms.
\[ u, u_i, u_d \in \mathbb{R}, \text{ from each of the error terms } e_p, e_i, e_d \in \mathbb{R}. \]

\[ u = f(r, y) = \lambda_p u_p + \lambda_i u_i + \lambda_d u_d \quad (2) \]

where

\[ u_p = K_p e_p, \quad u_i = K_i e_i, \quad u_d = K_d e_d \quad (3) \]

\[ e_i = \int_0^r (r - y) dt = \int_0^r e dt \equiv \frac{e(s)}{s} \quad (5) \]

\[ e_d = c \dot{r} - \dot{y} \equiv s (c r(s) - y(s)) \quad (6) \]

where \( K_p \in \mathbb{R} \) is the proportional gain, \( K_i \in \mathbb{R} \) is the integral gain, \( K_d \in \mathbb{R} \) is the derivative gain, \( u \in \mathbb{R} \) is the control action signal, \( e \in \mathbb{R} \) is the error signal, \( r \) is the set-point command, \( y \in \mathbb{R} \) is the measured plant state or output signal.

This critic formulation (2) differs from the conventional 2DOF PID definition which assumes that the three terms equally dominate the final output decision of the PID, that is \( \lambda_i, \lambda_p, \lambda_d \approx 1 \). Also, when \( \lambda_i \approx 0, \lambda_p, \lambda_d \neq 0 \), it reduces to a form of the classical 2DOF PD structure; and when \( \lambda_d \approx 0, \lambda_p, \lambda_i \neq 0 \), it reduces to a form of the classical 2DOF PI structure. As noted in [41], these are simplistic formulations of the natural error-based feedback control principle.

The parallel critic 2DOF PID structure defined above has two inherent closed loop transfer functions available in its structure using respective proportional and derivative set-point limiters or weights \( b \) and \( c \in [0, 1] \in \mathbb{R}^+ \) [1, 3, 10, 42–44]. The structure reduces to the common 1DOF error-feedback PID structure when the set-point weights are both unity. By the definition in [6], 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 (7a). The control output, \( u \) of this generalized continuous-form PID control function, has a distinct signal path to the reference set-point \( r \) and the output tracked signal \( y \) as illustrated in Figure 2. This kind of control structure exhibits the separation principle [10, 45] 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 [4, 46].

Simplifying (2) by assuming \( \lambda_p, \lambda_i, \lambda_d = 1 \) and rearranging, the result is a compact PID structure given by (7d) and illustrated in Figure 2.

\[ u = \left( K_p b + K_i \frac{s}{s} + K_d s c \right) r - \left( K_p + K_i \frac{s}{s} + K_d s \right) y \quad (7a) \]

\[ u = (K_p b + K_d s c) r + K_i s e - (K_p + K_d s) y \quad (7b) \]

\[ u = B(s) r(s) + A(s) e(s) - D(s) y(s) \quad (7c) \]

\[ u = B r + A e - D y \quad (7d) \]

where \( B, A, D \) are variables used to simplify the loop expressions in (7b), \( s \) is the complex Laplace operator (the \( s \) notation will sometimes be removed from transfer functions for simplicity).

The design problem then is to appropriately control the actual \( P(s) \) using the reduced PID expression (7d) without using the knowledge of an identified process model to make control predictions as is required by conventional PID control design methods.

### 3 Tuning Design

In this section, the Optimal Closed PID-Loop Model Predictive Control (OCPID-LMPC) tuning method is designed for the critic PID structure defined in section 2. The goal here is to adjust the PID parameters without using a dynamic process model to represent \( P(s) \).

#### 3.1 Closed PID-Loop

Tuning, without any loss of generality, implies a mathematically sound approach to adjusting the main gains \( K_p, K_i \) and \( K_d \) of the PID structure in order to achieve a specified closed-loop performance for \( P(s) \). In other words, finding the control gains that will push or pull a non-linear system mapping to a set-point [37].

\[ C = \frac{P(s)}{1 + D P(s)} \quad (8) \]

To achieve this goal, the first step is to use classical methods to obtain the transfer function \( T(s) \) of the overall closed loop system Figure 2. By performing a block reduction analysis (see (8), then signal-flow analysis (see Figure 3) using the Mason’s Gain Formula (9), where \( N \) is the number of forward paths from the input to output, \( P_i \) is the \( i_{th} \) forward path gain, \( \Delta \) is the signal flow determinant, and \( \Delta_i \) is the co-factor of \( \Delta \) along the \( i_{th} \) forward path. \( \Delta_i \) is obtained from \( \Delta \) by removing the loops not touching the \( i_{th} \) forward path in Figure 3. \( G(s) \), \( H(s) \), 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.

\[ T(s) = \frac{y(s)}{r(s)} = \frac{G(s)}{1 + G(s) H(s)} = \frac{\sum_{i=1}^{N} P_i \Delta_i}{\Delta} \quad (9) \]

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,
\[ \Delta = 1 - L_1 = 1 + AC, \ \Delta_1 = 1 \text{ and } \Delta_2 = 1. \] Applying (5), the expression (10) for the closed loop transfer function of Figure 2 is derived.

\[ T = \frac{C (A + B)}{1 + AC} = \frac{(A + B) P}{1 + (D + A) P} \]  

(10)

### 3.2 Stability Analysis

**Theorem 3.1.** The closed 2DOF PID-loop transfer function \( T(s) \), with the characteristic equation \( 1 + (D + A) P(s) \), is Hurwitz stable and achieves asymptotic performance as \( s \to 0 \), if the PID controller (2) defined by its three main gains, \( K_p \), \( K_i \) and \( K_d \), and critic weights \( \lambda_p \), \( \lambda_i \) and \( \lambda_d \), approximates the closed-loop response behaviour \( T(s) \) to a optimal second-order closed-loop model belief form \( T_m(s) \), with the characteristic equation \( s^2 + 2\zeta \omega_n s + \omega_n^2 \), in order to achieve asymptotic reference following of a command input trajectory \( y_m \).

**Proof.** To prove Theorem 3.1, we make the following restrictive assumptions for the closed PID-loop characteristic equation \( 1 + (D + A) P(s) \) given by the denominator of (10):

**Assumption 1.** Although all physical systems are nonlinear, the process \( P(s) \) is designed to operate in a linear or an approximately linear and bounded input-output range. The bounded-input bounded-output (BIBO) definition of stability states that: a system is stable if every bounded input produces a bounded output.

**Assumption 2.** If the dynamics of \( P(s) \) is BIBO stable and also affine in control according to Assumption 1. Then the unknown characteristic equation of \( P(s) \) is Hurwitz (all its eigenvalues lie in the open left-half of the complex s-plane).

**Assumption 3.** If Assumptions 1–2 are satisfied. As \( s \to 0 \), then the loop gain magnitude of the closed PID-loop controlled plant will dominate the unity of its characteristic equation, that is, \( |(D + A) P(s)| \gg 1 \).

**Remark.** Following Assumption 3, we note that the intuition, according to H.S. Black’s 1927 idea [44, 47, 48] of the negative feedback amplifier, to designing a successful negative 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 the process \( P(s) \).

If Assumptions 1–3 above hold, then, (10) can be approximated to (11). Where (11) which can be interpreted as the “common sense” or “closed loop predictive model” of the 2DOF PID control function (2) with respect to the output plant response \( y \). This is equivalent to (12), which is a general second-order optimal tuning function response. It also can be viewed as a general interpretation of the model-following adaptive control paradigm [7] for PIDs.

\[ \lim_{s \to 0} T \to T_m = \frac{A + B}{D + A} = \frac{1}{s^2} + \frac{bK_p}{K_d} s + \frac{K_i}{K_d} \]  

(11)

\[ y \approx y_m \to r \approx \frac{cs^2 + b2\zeta \omega_n s + \omega_n^2}{s^2 + 2\zeta \omega_n s + \omega_n^2} \]  

(12)

where \( \omega_n \) is the natural frequency, \( \zeta \) is the damping factor and \( y_m \) is the optimal output response achievable by the closed PID-loop and reference input to the closed PID-loop system.

Often, the zeros in the approximate closed loop transfer function (11) 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 (13) which transforms (11) to \( \hat{T}_m(s) \) defined by (14). This is equivalent to setting \( b \) and \( c \) to zero in (11).

\[ F = \frac{\hat{r}}{r} = \frac{\frac{K_i}{K_d}}{cs^2 + \frac{bK_p}{K_d} s + \frac{K_i}{K_d}} = \frac{1}{\omega_n^2} s^2 + \frac{2\zeta \omega_n s + 1}{s^2} \]  

(13)

\[ \hat{T}_m = \frac{\omega_n^2}{s^2 + 2\zeta \omega_n s + \omega_n^2} \]  

(14)

The expected closed loop PID eigenvalues (15) is obtained from the optimal characteristic denominator equation of (12), with \( \zeta \) taking a value of 0.707, for a optimal or near optimal [49] and robust [50] output response.

\[ s_{1,2} = -\zeta \omega_n \pm j\omega_n \sqrt{1 - \zeta^2} = -\kappa \omega_d \pm j\omega_d \]  

(15)

\[ \kappa = \frac{\omega_n}{\omega_d} = \frac{\zeta}{\sqrt{1 - \zeta^2}} \]  

(16)

where \( \omega_d \) is the damped natural frequency of oscillation or imaginary part of the conjugate complex poles, \( \kappa \) 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.

This completes the proof.

### 3.3 Setting \( K_d \) and \( K_i \)

Comparing (11) and (12), the result is the following nominal identities (17) and (18) for the respective gain and time-constant of the derivative and integral parts of the PID.

\[ K_d = f_d(K_p, \omega_n, \zeta) = K_p \frac{1}{2\zeta \omega_n}, \quad T_d = \frac{1}{2\zeta \omega_n} \]  

(17)

\[ K_i = f_i(K_p, \omega_n, \zeta) = \frac{K_p \omega_n}{2\zeta}, \quad T_i = \frac{2\zeta}{\omega_n} \]  

(18)

The set PID parameters (17) and (18) agrees with the result of [51] obtained by minimizing the quadratic cost function

\[ V = \int_0^\infty \left[ (r - y(t))^2 + \kappa_p^2 (u(t) - u(\infty))^2 \right] \]  

\[ = \int_0^\infty \left[ c(t)^2 + \Delta u(t)^2 \right] \]  

\[ u(\infty) = \frac{r}{\kappa_p}. \]  

(19)

(20)

where \( \kappa_p \) is the steady state process gain.

### 3.4 Closed PID-Loop Model Response

The expressions (17) and (18) set \( K_i \) and \( K_d \) but depends on \( K_p \) and \( \omega_n \). To progress towards a solution, in this section, we analyze the output state motion of closed loop predictive model (11) of the 2DOF-PID algorithm.
3.4.1 Normalized Model Output Response

The analytic time-domain response of the model can be obtained. A partial fraction analysis (PFA) of (14) leads to (23). Next, by taking the inverse Laplace transformation (ILT), we obtain expressions for the output response (21) and derivative (25) of the closed PID-loop model transfer function $T_m (s)$ to a unit step input, $r(s) = \frac{1}{s}$. The result is (24) and (28) respectively.

$$y_m (s) = \frac{cs^2 + 2b\zeta \omega_n s + \omega_n^2}{s(s^2 + 2\zeta \omega_n s + \omega_n^2)}$$  \hspace{1cm} (21)

$$y_m (t) = 1 + e^{-\zeta \omega_n t} \left[ \frac{(c-1) \cos (\omega_d t)}{2b - c - 1} \right]$$  \hspace{1cm} (24)

Also,

$$s y_m (s) = T_m (s) = \frac{cs^2 + 2b\zeta \omega_n s + \omega_n^2}{s^2 + 2\zeta \omega_n s + \omega_n^2}$$  \hspace{1cm} (25)

$$y_m (t) = 1 + e^{-\zeta \omega_n t} \left[ \frac{(b-c) 2\zeta \omega_n \cos (\omega_d t) + \omega_n^2 \left( \frac{(1-c)}{\omega_d} - \frac{(b-c) 2\zeta}{\omega_d} \right)}{\zeta \omega_n \sin (\omega_d t)} \right]$$  \hspace{1cm} (28)

If we let the normalized time $\omega_n t$, be $x = \zeta \omega_n t$, then the normalized model output response (24) and normalized model output response derivative (28) can be expressed compactly as (30) and (31) respectively.

Let $\kappa = \frac{\omega_n}{\omega_d} = \sqrt{1 - \zeta^2}$  \hspace{1cm} (29a)

$$\omega_d t = \frac{x}{\kappa}$$  \hspace{1cm} (29b)

$$y_m (x) = 1 + e^{-x} \left[ \frac{(c-1) \cos \left( \frac{x}{\kappa} \right)}{2b - c - 1} \right]$$  \hspace{1cm} (30)

$$\dot{y}_m (x) = c + e^{-x} \left[ \frac{(b-c) 2\zeta \cos \left( \frac{x}{\kappa} \right) + \left( \frac{(1-c)}{\kappa} - \frac{(b-c) 2\zeta}{\kappa} \right)}{\kappa \sin \left( \frac{x}{\kappa} \right)} \right]$$  \hspace{1cm} (31)

Further analyzing this state-space solution by assuming a constant or slowly varying set-point command signal $r$ leads to the derivation of the following interesting relationships (34)–(40).

$$y_m = x_2 + cr \quad \therefore \quad x_2 = y_m - cr$$  \hspace{1cm} (34)

$$\therefore \quad \dot{x}_{d,m} = -\dot{x}_2 \quad \therefore \quad \dot{\dot{x}}_2 = \ddot{x}_2$$  \hspace{1cm} (35)

$$\dot{x}_1 = \frac{\omega_n^2}{s} (x_2 + (1-c)r)$$  \hspace{1cm} (36)

$$\therefore \quad e_m = r - y_m = \frac{x_1}{\omega_n^2}$$  \hspace{1cm} (37)

$$\dot{x}_2 = x_1 + 2\zeta \omega_n (x_2 + (b-c)r) \quad \therefore \quad e_{p,m} = br - y_m = \frac{\dot{x}_2 - x_1}{2\zeta \omega_n}$$  \hspace{1cm} (39)

leading to

$$u_m = K_p e_{p,m} + K_i e_{i,m} + K_d e_{d,m}$$  \hspace{1cm} (40)

and

$$e_{u,m} = u(\infty) - u_m$$  \hspace{1cm} (41)

The industrial form of the PID favours the I-PD structural form ($b = 0, c = 0$) equivalent to a integral plus state feedback controller, meanwhile a more general form favours the PI-D form ($b = 1, c = 0$). With respect to these common PID forms, using the derived relationships from the state-space solution (33), we graphically observe the model output and error response that a PID controlled closed loop plant should follow. This is illustrated in the two example normalized plots in Figure 4 and Figure 5.

It is observed that the response of the closed 2DOF-PID loop model is affected by the choice of $b$ and $c$. Changing values of $b$ and $c$ lead to variations in the normalized peak times and settling times of $T_m (s)$ to the closed PID-loop model response which affects the PID’s model input-output surface. The plot of the input-output surface of the PID model, as illustrated in Figure 4h and Figure 5h, reveals a convex shaped mapping curve.

In the next section, in order to set the model natural frequency $\omega_n$, we analyze the normalized plots of the model output response and its first derivative. The first undershoot area of the model output response is considered to obtain its approximate normalized settling time. For the normalized peak time, the first point at which the derivative of the model output response is zero is considered. The result is Table 1 and 2.

3.5 Setting $\omega_n$ by fuzzy inference of the closed PID-loop response

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 [52], leads to an intuitive choice of the settling time plus delay time of the open-loop physical process in setting the model closed-loop natural frequency $\omega_n$. 

\[ A = \begin{bmatrix} -\omega_n^2 & 1 \\ -2\zeta \omega_n & 0 \end{bmatrix}, \quad B = \begin{bmatrix} \omega_n^2 (1-c) \\ 2\zeta \omega_n (b-c) \end{bmatrix} \]  \hspace{1cm} (33)
Table 1. Normalized Peak Time: Closed PID-Loop Model’s Fuzzy Membership Function Centers, \( c \) \(^1\)

| c     | 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 | 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 |

\(^1\) \( d = 0.01 \)

Table 2. Normalized Settling Time: Closed PID-Loop Model’s Fuzzy Membership Function Centers, \( c \) \(^1\)

| c     | 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 |

\(^1\) \( d = 2.22 \) if \( b = 0 \) or \( b = 1 \), otherwise \( d = 2.08 \)

From the classical time-domain relation \( \omega_n = \frac{x}{t_n} \), the desired closed loop natural frequency \( \omega_n \) at the normalized settling time \( x = x_s \) with respect to the operating settling time \( t \) of the actual open-loop process can be calculated. Alternatively, the desired \( \omega_n \) may also be calculated using the normalized peak time \( x = x_{pk} \) value, with respect to the operating open-loop peak time value \( t_{pk} \).

\[ \omega_n = \frac{x_s}{t_n} \text{ or } \omega_n = \frac{x_{pk}}{t_{pk} \sqrt{1 - x^2}} \]  

(42)

To compute \( \omega_n \) using (42), our approach is to encode the fuzzy behaviour of the optimal response of the closed PID-loop model with changing \( b \) and \( c \) as a fuzzy inference system (FIS), defined by (44), and synthesized as a fuzzy basis function (FBF) [8] defined in (45). This method is used 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 3, while the input-output nonlinear mapping surface of the FIS is shown in Figure 6.

**Definition 3.1.** For input \( x = [b \ c]^T \), the Type-1 FIS that maps to \( y = [x_s \ x_{pk}]^T \) is represented as the fuzzy basis

| Table 3. Type-1 FIS Design Choice |
|----------------------------------|
| **FIS Type** | Takagi-Sugeno-Kang (TSK) |
| **Input(s)** | \( b \) and \( c \) : \( p = 2 \) |
| **Antecedent Fuzzifier** | Singletons |
| **Universe** | \( 0 \) – \( 1 \) |
| **Output(s)** | \( x_{pk} \) or \( x_{ts} \) |
| **Consequent Fuzzifier** | Singleton or Non-Singleton |
| **Universe** | \( 2 - 6 \) or \( 0 - 20 \) |
| **T-norm** | Product |
| **T-conorm** | Max |
| **MF (Parameterized)** | closed \( n \)-logistic sigmoid, see (45) |
| **MF Parameters** | Pre-specified, see Table 1 and 2 |
| **Number of Rules** | \( M = 121 \) for each output |
(a) | (b) | (c) | (d) | (e) | (f) | (g) | (h)

Figure 4. Closed PID-loop model’s state space response analysis, when \( b = 1, \ c = 0; (4a) output \( y_m \) and error \( e_m \), (4b) output rate \( \dot{y}_m \) and error rate \( \dot{e}_m \), (4c) integral error \( e_{i,m} \), (4d) proportional \( e_{p,m} \), (4e) control output \( u_m \), (4f) output phase plot, (4g) error phase plot, and (4h) control output surface map.

(a) | (b) | (c) | (d) | (e) | (f) | (g) | (h)

Figure 5. Closed PID-loop model’s state space response analysis, when \( b = 0, \ c = 0; (5a) output \( y_m \) and error \( e_m \), (5b) output rate \( \dot{y}_m \) and error rate \( \dot{e}_m \), (5c) integral error \( e_{i,m} \), (5d) proportional \( e_{p,m} \), (5e) control output \( u_m \), (5f) output phase plot, (5g) error phase plot, and (5h) control output surface map.

function expansion:

\[
y(x) = \sum_{l=1}^{M} c_l \phi^l_j(x) \tag{43}
\]

\[
\phi^l_j(x) = \frac{\prod_{i=1}^{l} \mu^l_{F_i}(x_i)}{\sum_{i=1}^{M} \prod_{i=1}^{l} \mu^l_{F_i}(x_i)} \tag{44}
\]

\[
\mu^l_{F_i}(x_i) = \begin{cases} 
\mu_L(x) = \text{nlsig}^- \left( x; \bar{c}_L, \bar{d}_L \right) ; x < \frac{\bar{c}_L + \bar{d}_L}{2} \\
\mu_R(x) = \text{nlsig}^+ \left( x; \bar{c}_R, \bar{d}_R \right) ; x > \frac{\bar{c}_R + \bar{d}_R}{2}
\end{cases} \tag{45}
\]

Figure 6. FIS Output surface defined by (44) for the closed PID-loop model.

For \( \text{nlsig}^- \), the following constrains hold \( x^-_{\min} = \bar{c}_L - \bar{d}_L \), \( x^-_{\max} = \bar{c}_L \), \( \bar{c}_L > x^-_{\min} \) and \( x^+_{\min} = \bar{c}_R \), \( x^+_{\max} = \bar{c}_R + \bar{d}_R \). Also, \( y_{\max} = 1, y_{\min} = 0 \).

Definition 3.2. The \( n \)-logistic sigmoid function, where \( \delta \in \mathbb{R}^{n+1} \) and \( \kappa_{x}, \kappa_{y} \in \mathbb{R}^{(n+1)+1} \) with \( \lambda = 6 \) as a standard default value is defined as:

\[
y = \text{nlsig}^\pm \left( x; x_{\min}^+, x_{\max}^-, n, \lambda \right) = \kappa_{y,1} + \sum_{i=1}^{n} \kappa_{y,i+1} - \kappa_{y,i} \tag{46}
\]

The following holds:

\[
\begin{aligned}
\lim_{x \to x^-_{\max}} \text{nlsig}^- (x) &= y_{\max} \\
\lim_{x \to x^-_{\min}} \text{nlsig}^- (x) &= y_{\min} \\
\lim_{x \to x^+_{\min}} \text{nlsig}^+ (x) &= y_{\max} \\
\lim_{x \to x^+_{\max}} \text{nlsig}^+ (x) &= y_{\min}
\end{aligned} \tag{47}
\]

\[
\Delta_x = \frac{x_{\max} - x_{\min}}{n}, \quad \Delta_y = \frac{y_{\max} - y_{\min}}{n} \tag{48}
\]

\[
\alpha = \frac{\kappa_{x,i+1} - \kappa_{y,i}}{\left(2 - \kappa_{x,i+1} - \kappa_{y,i} \right)} \tag{49}
\]

\[
\delta_i = \kappa_{x,i+1} - \kappa_{y,i}, \quad i = 1, \ldots, n \tag{50}
\]

\[
\kappa_{x,i+1} = \kappa_{x,i} + \frac{\Delta_x}{\Delta_y}, \quad \kappa_{y,i+1} = \kappa_{y,i} + \Delta_y \tag{51}
\]

where \( \kappa_{x} = [\kappa_{x,1}, \ldots, \kappa_{x,i+1}] \), \( \kappa_{x,i} = x_{\min} \), \( \kappa_{x,n+1} = x_{\max} \) and \( \kappa_{y} = [\kappa_{y,1}, \ldots, \kappa_{y,i+1}] \), \( \kappa_{y,i} = y_{\min} \), \( \kappa_{y,n+1} = y_{\max} \).

3.6 Setting \( K_p \)

The remaining PID parameter to set is the proportional gain \( K_p \). Practically, the power limitations (noise and saturation) of the hardware actuator, which parses the controller’s output to effect a process output, limits how high \( K_p \) can be set. From (17) and (18), it is clear that \( K_p \) is the PID’s global gain parameter upon which the two other gain parameters are dependent. The inquiry then is: how to determine a stabilizing real-valued \( K_p \) that will effectively regulate \( P(s) \) and constrain the actuator’s energy within limits without resulting to the use of a process model? In this paper, we propose an hybrid solution.

First, our approach is to set \( y_m \) as the input to the PID and tune \( K_p \) as an adaptive function of the resulting error \( y_m - y \). In standard controllable state form, the closed PID-loop model
is
\[ \dot{x}_m = A_m x_m + B_m r \]
\[ y = C_m x_m + D_m r \]  

(52)

\[ A_m = \begin{bmatrix} 0 & 1 \\ -\omega_n^2 & -2\zeta\omega_n \end{bmatrix} \]
\[ B_m = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \]  

(53)

\[ C_m = \begin{bmatrix} \omega_n^2 (1 - c) \\ 2\zeta\omega_n (b - c) \end{bmatrix} \]
\[ D_m = [c] \]

If we assume the control law is
\[ u = u_p = K_p \dot{e} = K_p (b x_m - x_p) \]  

(54)

Assuming an attenuated effect of plant uncertainty, the unknown plant state dynamics (1) is \( \dot{x}_p = A x_p + B u \). The tracking error is \( e = x_m - x \) and its derivative is \( \dot{e} = \dot{x}_m - \dot{x}_p \). For perfect closed-loop model matching or following, \( x_p = x_m \) and \( x_m = r \), then we have
\[ \dot{x}_p = (A - BK_p) x_p + BK_p b x_m \]  

(55)

Comparing (55) with (52), we obtain the following identities
\[ A_m = A - BK_p (\infty) \]
\[ B_m = BK_p (\infty) \]  

(56)

\[ : A = A_m + \frac{B_m}{b} \]  

(57)

The tracking error becomes
\[ \dot{e} = A_m e + B (K_p (\infty) - K_p) e \]  

(58)

As a candidate lyapunov function \( V \), we choose a quadratic function of the error and \( K_p \),
\[ V(e, K_p) = \frac{\gamma e^2}{2} + B (K_p (\infty) - K_p)^2 \]  

(59)

Then its derivative \( \dot{V} \) is
\[ \dot{V} = \gamma \dot{e} e + B (K_p (\infty) - K_p) \dot{K}_p \]  

(60)

\[ \dot{V} = \gamma A_m e^2 + B (K_p (\infty) - K_p) \left( \gamma e e_p + \dot{K}_p \right) \]  

(61)

The parameter update law can be selected as
\[ \dot{K}_p = -\gamma e e_p, \quad \gamma = 0.001 \]
if \( b = 1, \quad \dot{K}_p = -\gamma e^2 \]
\[ K_p = K_p - \dot{K}_p \]  

(62)

(63)

Using Barbalat’s lemma it is easy to see that \( e \) asymptotically converges to zero as time goes to infinity. This implies that the controlled plant output \( y \) tends to the closed-PID loop model output \( y_m \). An initial value for \( K_p \) can be taken as 0.01, where \( \gamma \) is also a very small positive constant.

However, there is always a form of delay, even if negligible, in most processes. When this delay gets larger, the solution (62) obtained from the approach above cannot alone guarantee the stability of the closed-loop system. To solve this problem, we apply an augmented approach.

To set \( K_p \) within appropriate bounds, our approach is to use a nonlinear sum of error, \( e \) and the control action \( u \) using the forward form of the newly formulated n-logistic sigmoid function \( nlsig^- \) in Definition (3.2), with, \( y_{\text{max}} = x_{\text{max}} = k_{\text{lim}} \), and \( y_{\text{min}} = x_{\text{min}} = 0 \). We wish to constrain \( K_p \) to a value between 0 and \( k_{\text{lim}} \). However, the maximum possible positive stabilizing value \( k_{\text{lim}} \) is unknown. To obtain an approximate value for \( k_{\text{lim}} \), we analyse a PID-controlled normalized first-order plus dead time model with increasing values of normalized delay time \( L \). The obtained data values of \( L \) and \( k_{\text{lim}} \) that leads to a stabilizing \( K_p \) value is fitted to a normalized rational function (68) multiplied by the ratio of delayed settling-time to the settling-time without delay. The overall solution (64) is expressed below.

\[ K_p = f_p (e, u, k_{\text{lim}}) = nlsig^- (k_0; 0, k_{\text{lim}}, n_p, \lambda_k) \]  

(64)

where:
\[ k_0 = nlsig^- (e (t); -k_{\text{lim}}, k_{\text{lim}}, n_e, \lambda_k) \]
\[ + nlsig^- (u (t); -k_{\text{lim}}, k_{\text{lim}}, n_u, \lambda_k) \]
\[ n_p = 16, \quad n_e = n_u = 1, \quad \lambda_k = 0.1 \]
\[ k_{\text{lim}} = \alpha n_s \left( \frac{L + t_s}{t_s} \right), \quad k_{\text{lim}}, \alpha \in \mathbb{R}^+ \]  

(67)

By default, \( \alpha = 1 \)
\[ \kappa_p = \frac{p_1 L^2 + p_2 L + p_2}{L^2 + q_1 L + q_2} \]
\[ p_1 = 0.05132 \]
\[ p_2 = 0.2041 \]
\[ q_1 = 1.538 \]
\[ q_2 = 0.5864 \]
\[ L = \frac{L - 5.936}{8.771} \]  

(69)

When \( t > L \)
\[ K_p = K_p - \dot{K}_p \]

The sigmoidal limiting function (64) is formulated in such a way that it estimates and constrains the maximum value of \( K_p \) with respect to a free variable hyper-parameter \( \alpha \) and the presence of a delay \( L \) in the settling time.

**Remark.** The realization of Theorem 3.1 for different physical systems depends on the value of the constrained global PID gain parameter \( K_p \) defined with (64). However, provided the Assumptions (1)–(3) are satisfied, the output step response of a PID controlled system should asymptotically converge towards the stable response of the closed PID-loop model response.

### 4 SAD Tuning Algorithm

Based on the optimal closed PID-loop model predictive control method proposed in section 3, in this section, we present an automatic tuning algorithm named “SAD”.

For a periodic triggered control framework, where \( T \leq 0.01 \) seconds is the sampling time, the SAD algorithm is outlined in Algorithm 1.

The derivative filter time-constant \( T_f \) is usually used to make the measured error derivative causal or to attenuate the noise associated with it.

### 5 Simulation Results and Discussion

Simulation results of the proposed method, benchmarked against other selected methods in literature will be presented and discussed in this section.
Algorithm 1 SAD Tuning Algorithm

Require: \( T, L, t_s, \zeta, b, c, e(t), u(t) \)
1: \( t_s \leftarrow t_s - L \quad \triangleright \text{settling time without delay.} \)
2: \( \tau_L \leftarrow (L + \tau_c)/t_s \)
3: \( [x_s, x_{pk}] \leftarrow f_{FIS}(b, c), \quad \omega_n \leftarrow f_{\omega_n}(\zeta, x_s, t_s) \)
4: \( K_p \leftarrow f_p(e(t), u(t), K_{ini}), \quad \lambda_p \leftarrow 1 \)
5: \( K_i \leftarrow f_i(K_p, \omega_n, \zeta) \)
6: if \( L > T \) then
7: \( \lambda_i \leftarrow \lambda^1_i \)
8: else
9: \( \lambda_i \leftarrow \lambda^2_i \quad \triangleright \lambda^2_i \geq \lambda^1_i > 0. \)
10: end if
11: \( \lambda_d \leftarrow \lambda^3_d, \quad \lambda^1_d \leftarrow L \times (x_s - x_{pk}) \)
12: end if
13: \( T_f \leftarrow f(K_p) \leftarrow 1/\kappa_p^2 \)

Table 4. Tuning Rules$^1$.

| Method               | \( K_p \)            | \( K_i \)            | \( K_d \)            |
|----------------------|----------------------|----------------------|----------------------|
| CHR, Chiens, Hrones and Reswick ($0\%$) | $0.6 \frac{1}{\kappa_p L}$ | $0.6 \frac{2m}{L}$ | $0.3 \frac{1}{L} m$ |
| SAD, Somefun, Akingbade and Dahansi | \( f_p(\cdot) \) | \( f_i(\cdot) \) | \( f_d(\cdot) \) |

$^1$ 0% set-point regulation. Where \( \tau \) is process time constant and \( \kappa_p \) is steady-state gain of the process.

We have developed a method for optimizing the step response of a closed-loop system consisting of a 2DOF PID controller \((b = 1 \text{ and } c = 0)\) and the unknown dynamics of a LTI SISO plant. Compared to the Z-N methods, instead of using a step-response experiment to obtain a first-order plus dead time plant model estimate, we use this experiment to estimate the delayed settling time of the plant as its characteristic linear dynamic behaviour. The effect of this is that we restrict the use of identified process models as representations of the true process for simulation purposes and not for the control design.

As standard performance measures, the quadratic integral performance error indices in form of \( J_{ise} \) and \( J_{isu} \) are good measures of set-point regulation. Where: \( e(t) \) is the error due to a step change in the set-point command signal, \( u(t) \) is the control input from the controller, \( u(\infty) \) is the optimal steady-state control input corresponding to the desired output. Also, an instinctive requirement for modern PID tuning methods is to ensure specified closed loop performance in terms of the maximum overshoot and settling time [2,3,18,28]. Therefore, the controller tuning performance is also assessed using the settling time and percentage maximum overshoot indices of the closed-loop with respect to a unit-step input.

\[
J_{ise} = \int_0^\infty e(t)^2, \quad J_{isu} = \int_0^\infty (u(\infty) - u(t))^2
\]

5.1 First Order Plus Dead Time Model

Consider a representative first-order plus dead time (FOPDT) process model for a dc motor plant, where \( \kappa_p = 1, \tau = 1, L = 2 \). The controllability ratio of this plant is 0.67. This implies a lag dominated (difficult to control) plant.

Applying the proposed SAD algorithm in comparison with the CHR method, JJ-ST method and the AMIGOS method; Figure 9 shows the normalized process response to a unit-step command, and Table 5 shows the steady-state controller parameters and performance.

It is deduced from the performance indices in Tables 5 that all four PID tuning methods achieve set-point regulation successfully. The AMIGO Response is smooth, no overshoot but most sluggish. Using the CHR method, the output response has the highest percentage maximum overshoot, while for the JJ-ST method, a more oscillatory response is observed. Compared to
these methods, the overall best performance is obtained by the OCPID-LMPC method through the SAD algorithm which provides the best balance between the closed-loop settling time and maximum overshoot of the output response. It, therefore, can be inferred from the above results that the OCPID-LMPC method is very capable of giving good control performance even in the presence of dead-time for this class of systems, compared to these methods, the overall best performance is obtained by the OCPID-LMPC method through the SAD algorithm which provides the best balance between the closed-loop settling time and maximum overshoot of the output response. It, therefore, can be inferred from the above results that the OCPID-LMPC method is very capable of giving good control performance even in the presence of dead-time for this class of systems, compared to the other methods that use the knowledge of a formal process model representation of the actual plant.

5.2 Speed Control of a DC Motor

Next we consider a more realistic representation (oscillatory plant model) of an actual permanent magnet dc-motor plant with specifications shown in Table 6.

\[
P_1(s) = \frac{1}{(1 + \frac{1}{\tau_0}e^{-\frac{s}{\lambda}})}
\]

where:

\[
P_2(s) = \frac{b_0}{s^2 + a_1 s + a_2}
\]

Figure 8. Responses to a unit step command for \(P_1(s)\): Fig. (9a.) Control signal output, Fig. (9b.) Controlled output response. SAD (solid red; \(\alpha = 1\)), \(\lambda_1 = 1.25\), \(\lambda_d = 11.08\), optimal closed PID-loop model response (dashed red), JJ-ST (dash-dotted gray), CHR (solid black), AMIGO (dashed gray).

Table 5. Controller Parameters and Performance Measures\(^1\) for the Plant

| Name | Description | Value |
|------|-------------|-------|
| \(J_m\) | moment of inertia, Kg.m\(^2\) | 0.2752 |
| \(B_m\) | viscous Friction constant, N.m.s | 0.3922 |
| \(K_t\) | motor torque constant, N.m/A | 1.882 |
| \(K_b\) | electromotive force constant, V.s/rad | 1.185 |
| \(R_a\) | armature resistance, \(\Omega\) | 0.1557 |
| \(L_a\) | armature inductance, H | 0.00082 |
| \(n_g\) | gear ratio | 1.94 |
| \(t_o^p\) | process settling time | 0.02 |

\(a_1 = \frac{R_a J_m + L_a * B_m}{J_m L_a}\), \(a_2 = \frac{R_a B_m + K_t K_b}{J_m L_a}\) \hspace{1cm} (72)

\(b_0 = \frac{1}{n_g^2 J_m L_a} K_t\) \hspace{1cm} (73)

The settling time of the process output (speed) is 0.02 seconds. A sensor noise of \(\pm 20\%\) and a state noise of \(\pm 1\%\) modelled as a gaussian process is assumed for a unit-step response command. Also, a unit step disturbance was introduced at 0.2 seconds. Figure 9 illustrates the performance of the OCPID-LMPC method and varying PID gains, with respect to the set-point regulation of the dc motor’s process output (speed). The resultant performance indices is \(J_{se} = 0.032\).

It can be concluded from the performance measures in this section that the OCPID-LMPC method in relation to the speed control of the dc motor plant gives a robust closed loop system in the presence of bounded uncertainties acting on the measured output process.

5.3 Discussions

Many sophisticated estimation techniques exist for process or system identification [54]. In itself process identification helps control designers understand the limits and the input-output behaviour of the process. The proposed method, however, uses a very simplistic form of low-level process identification that can be quite easily carried out based on input-output data to estimate the process gain, delay time and settling time of the process, without formulating a process model. The use of process models is used in its primary application to simulate an approximate behaviour of the actual plant and test the performance of the tuned PID controller [55].
Although PID design is regarded by many as an established area, according to [18, 56–59], engineering practice is persistently demanding for PID control design methods that simultaneously guarantee the metrics of maximum overshoot and settling time. Researchers [17, 60–62] agree that a satisfactory controller can be a PID if its parameter gain space is well tuned or exploited. For all of the test cases provided above, the output response of the process follows the optimal closed-PID loop output response with respect to the correct estimations of the final settling time and time-delay of the open loop process.

For control tasks involving a process variable like the speed or position of a dc motor, regulating such around their operating regions can become less complex and more automatic. Good or near optimal control gains can be computed automatically without the use of uncertain exaggerated process models, hence the time and effort taken to compute optimal control actions is reduced significantly. Further, it is straightforward to see that this proposed OCPID-LMPC method has better and comparable control performance when properly applied. Therefore, this method has promising use in place of the popular model-based PID tuning methods, as a first try for tuning PIDs in dc motor applications.

6 Conclusions

In this paper, we first reformulated the PID structure to include critic gains which penalize each of its three terms. Next the designed OCPID-LMPC adaptive tuning method was proposed. The OCPID-LMPC method is a flexible framework as different automatic algorithms can be designed based on it. The resulting tuning algorithm in this paper, SAD, is fast and quite easy to implement as a software library, so that industry engineers and non-expert users can flexibly employ PIDs to achieve good quality of control performance with efficient expenditure of control energy compared to other established PID tuning methods. As proof of concept, simulation results show that this process model-free PID tuning method can approximate the optimal solution for a specific class of physical processes. For performance validation, the method was demonstrated on some benchmark systems and compared with some key PID tuning methods in the literature. From performance results, the proposed method outperforms: the CHR method in minimizing the quadratic cost function; and the convex-concave optimization in minimizing the integral absolute error. Hence we conclude that this systematic approach to PID design (tuning) has potential for applicability to many PID controllable plants even when noisy and dead-time dominated, with a promising industrial application being the robust position and speed control of electric motor drives and in motion control.

REFERENCES

[1] 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.

[2] 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.

[3] M. Viteckova and A. Vitecek, “2DOF Controller Tuning,” International Journal of Engineering Research in Africa, vol. 18, pp. 57–64, 2015.
[32] B. Hekimoğlu, “Optimal Tuning of Fractional Order PID Controller for DC Motor Speed Control via Chaotic Atom Search Optimization Algorithm,” IEEE Access, vol. 7, pp. 38 100–38 114, 2019.

[33] R. K. Mandava and P. R. Vundavilli, “An optimal PID controller for a biped robot walking on flat terrain using MCIWO algorithms,” Evolutionary Intelligence, vol. 12, no. 1, pp. 33–48, Mar. 2019.

[34] A. Almabrok, M. Psarakis, and A. Dounis, “Fast Tuning of the PID Controller in An HVAC System Using the Big Bang–Big Crunch Algorithm and FPGA Technology,” Algorithms, vol. 11, no. 10, p. 146, Oct. 2018.

[35] A. Taeib and A. Chaari, “Tuning optimal PID controller,” International Journal of Modelling, Identification and Control, vol. 23, no. 2, pp. 140–147, Jan. 2015.

[36] M. Elwin, K. M. Lynch, and N. Marchuk, Embedded Computing and Mechatronics with the PIC32 Microcontroller, 1st ed. Newnes is an imprint of Elsevier, 2015.

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

[38] T. Ray Chaudhuri, L. G. C. Hamey, and R. D. Bell, “From conventional control to autonomous intelligent methods,” IFAC Proceedings Volumes, vol. 23, no. 8, Part 2, pp. 443–448, Aug. 1990.

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

[40] A. A. Mahfouz, A. A. Aly, and F. A. Salem, “Mechatronics Design of a Mobile Robot System,” International Journal of Intelligent Systems and Applications (IJSIA), vol. 5, no. 3, p. 23.

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

[42] K. J. Åström and T. Hägglund, “Benchmark Systems for PID Control,” IFAC Proceedings Volumes, vol. 33, no. 4, pp. 165–166, Apr. 2000.

[43] 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.

[44] K. J. Åström and T. Hägglund, Advanced PID Control. ISA Instrumentation, Systems and Automation Society, 2006.

[45] 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.

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