

# Three-time scale multi-objective optimal PID control with filter design

Muhammad Aafaque<sup>1</sup>  | Almuatazbellah Boker<sup>2</sup> | Yousef Sardahi<sup>3</sup>

<sup>1</sup>R&D Center, Karl Dungs GmbH & Co. KG, Urbach, Germany

<sup>2</sup>Bradley Department of Electrical and Computer Engineering, Virginia Tech, Blacksburg, Virginia, USA

<sup>3</sup>Department of Mechanical Engineering, Marshall University, Huntington, West Virginia, USA

## Correspondence

Muhammad Aafaque, R&D Center, Karl Dungs GmbH & Co. KG, D-73660 Urbach, Baden-Wuerttemberg, Germany.  
Email: engr\_aafaque@hotmail.com

## Abstract

This paper presents the design of a multi-objective PID (proportional, integral, and derivative) controller in three time scales for a system with load disturbances and sensor noise. The key to this design method is to divide the problem into three time scales by following a singular perturbation approach, which allows for the optimization of fewer parameters in each time scale instead of all the three PID gains at once, hence less computation and design effort. The optimization objectives are the minimization of the overshoot and peak time and the maximization of the closed-loop system's ability to reject noise and load disturbances. The impact of the integral action and filter on the optimal results is investigated by tuning the singular perturbation parameters. The obtained results show that the performance of the closed-loop system recovers the optimal solution, which is based on the fast subsystem, as the singular perturbation parameters get sufficiently small.

## KEYWORDS

load disturbance, measurement noise, modified PID, multi-objective optimization, multitime scale

## 1 | INTRODUCTION

Feedback control design usually involves meeting multiple performance objectives which might be conflicting. It is sometimes not possible to simultaneously maximize and/or minimize the performance objectives. This paradox is especially clear when it comes to designing the parameters of the proportional, integral and derivative (PID) controller. This is an important problem to solve as the PID controller is one of the popular feedback controllers that is widely used in the industry thanks to its simplicity and efficiency [1, 2].

Both the deterministic and stochastic frameworks have been proposed for the tuning of PID controllers [3]. The most commonly used conventional methods for tuning the PID controller are trial and error method,

relay tuning method, pole placement [4], and minimum variance techniques [5], Ziegler–Nichols step response method, Ziegler–Nichols frequency response method [6], and Cohen–Coon method (1953) [1]. The model of the system is not necessarily required to determine the gains of PID controller in such conventional methods which makes them advantageous over others [7]. However, such methods like Ziegler–Nichols may often lead to oscillations and big overshoot in the outputs when subjected to set-point changes as modeling uncertainties are involved despite good disturbance rejection [8]. Moreover, these methods are not intended to handle more than one design requirement in multi-objective settings and cannot deal with more than one cost function, for example, minimum rise time, overshoot, and regulation time at the same time, a new optimal design of the PID controller is required [9]. A

number of attempts have been made to solve the tuning problem of gains of the PID controller, for example, in Ossareh et al. [10], an internal model control (IMC) based approach is used to tune the gains of the PID controller for the boost control system of the turbocharged engine using relay feedback experiment. In Matusšů et al. [11], a robust PI controller is designed for interval plants using worst-case gain and phase margin specifications in the presence of multiple crossover frequencies, and the oblique wing aircraft is selected as an interval plant to demonstrate the results of the proposed design. In Barros et al. [12], a technique is proposed for designing robust PID controller for a class of linear time-invariant single-input single-output systems using a linear programming approach for optimizing performance subject to robustness constraints, in which the reference loop gain transfer function is shaped to form a convex region on the Nyquist diagram, where convex region is approximated by a set of lines to formulate the optimization problem.

Stochastic framework has gotten a lot of attention in solving the control problems [13], especially the multi-objective control problems [14]. Evolutionary algorithm (EA) is used in Li et al. [15] to optimize the gains of PD controller for pneumatic rotary actuator servo system with friction compensation, and the results are compared with that of differential evolution algorithm (DEA) and genetic algorithm (GA). Nondominated sorting genetic algorithm (NSGA-II) is used for optimization of PID parameters for attitude stabilization of hovering quadcopter in Yoon and Doh [16], where a neural network algorithm, long short-term memory (LSTM), was used to evaluate the flight motions based on obtained PID parameters.

Feedback controllers come with a disadvantage of injecting measurement noise in the system which causes unwanted oscillations and, in consequence, wear and tear which may lead to possible breakdown [17]. As the measurement noise is usually dominated by higher frequencies [17], a low-pass filtering is required for the derivative part of the PID controller to limit the high-frequency gain [18]. The work by Sardahi and Boker [19] proposed a second-order PID controller, which requires tuning of four parameters including the derivative filter factor, and it was shown in that the derivative filter factor must be taken into account while optimizing the PID controller. This was a departure from the practice of designing commercial controllers as the filter gain is usually kept constant or totally ignored due to its noise amplification characteristics [20], especially when input disturbances are considered. The flip side is that this leads to a four-parameter design problem instead of three-parameter one, causing stability analysis more complex and tuning process more difficult

or computationally demanding. However, using derivative filter factor in the PID controller improves the stability of closed-loop system, transient response, and load disturbance rejection [20]. The results of Sardahi and Boker [19] confirm this claim by following multi-objective optimization framework. Despite all this progress, improvement is still needed when it comes to design complexity and computational cost.

The desired output transient performances, and their insensitivity to plant parameter variations and unknown external disturbances, are ensured for full-order closed-loop nonlinear system by imposing sufficiently large mode separation rate between fast and slow modes [21]. In Lorenzetti and Weiss [22], an antiwindup PI controller with a saturating integrator is proposed for a single-input single-output stable nonlinear system, and stability analysis is performed using a singular perturbation method by dividing the system into the slow and fast subsystem. However, this work does not consider measurement noise, and neither filter is employed nor PI is tuned using any optimization techniques. In Yurkevich [23], the tracking problem for multiple-input multiple-output (MIMO) nonlinear systems is addressed by multivariable PI/PID controller based on time scale separation (singular perturbation) technique, and author claims near-perfect rejection of nonlinearities, unknown external disturbances, and loop interactions due to increased degree of time scale separation, but this work also does not take measurement noise into consideration, and the parameters of the PI/PID controller are computed analytically instead of using any optimization technique.

In Pan et al. [24], a linear PID controller is used for tracking control of robotic manipulators actuated by compliant actuators and formulating the control problem into three-time scale singular perturbation method where slow time scale is included at the rigid dynamics, actual fast time scale is included at the actuator dynamics, and one virtual fast time scale is included at controller dynamics. This approach not only provides semiglobal practical exponential stability using proper choice of control parameters but also is structurally simple and model-free resulting in low implementation cost and robust against external disturbances and parameter variations. Though this work utilizes the three-time scale technique to design PID controller, yet it does not include the filter in the derivative part of the controller to counter the measurement noise.

This work proposes a method that designs the PID controller with a measurement derivative filter through solving a multi-objective optimization problem (MOP) in a systematic and computationally efficient way. It avoids solving the optimization problem in one shot thus improving the result reported in Sardahi and Boker [19], where

four tuning parameters need to be optimized at the same time. The key idea in our work is to exploit the fact that the integrator dynamics are typically slower than the system dynamics while the filter dynamics are typically faster. This leads to modeling the closed-loop system in a three-time scale fashion. Singular perturbation theory is then employed to decompose the closed-loop fourth-order system into three smaller-dimensional subsystems. This considerably simplifies the control design and allows for a targeted design thanks to the decoupling of the system states and parameters. Once the closed-loop system is decomposed into lower dimensional submodel, we design the gains of the PID controller using the NSGA-II, which is one of the most widely used multi-objective optimization algorithms [25], through minimizing the closed-loop response overshoot, peak time and maximizing the ability of the system to reject external load disturbances and measurement noises. The main idea here is to account for the effect of the derivative filter on these conflicting design goals and present that to the decision-maker before the real implementation takes place. We show that the desired performance of the closed-loop system that resulted from the decomposed system can be recovered after implementation by making the integrator sufficiently small and the filter sufficiently fast.

The remainder of this paper is organized as follows. Section 4 introduces the concept and the definition of the multi-objective optimization. The general formulation of the proposed control problem is proposed in Section 2. Section 5 introduces a numerical example, the multi-objective control design, and the results of the optimization process. Finally, the paper is concluded in Section 6.

## 2 | PROBLEM FORMULATION

Consider a plant  $G_P$  described by

$$\dot{x}_1 = x_2, \quad (1)$$

$$\dot{x}_2 = a_1x_1 + a_2x_2 + b(u + d), \quad (2)$$

$$y = x_1 + n, \quad (3)$$

where  $x = [x_1 \ x_2]^T \in \mathbb{R}^2$  is the vector of the system states,  $a_1, a_2$  are systems parameters,  $u \in \mathbb{R}$  is the control input and  $b \neq 0$ ,  $y \in \mathbb{R}$  is the measured output,  $d$  is the unknown external disturbance and  $n$  is a measurement (white) noise.

The objective of this work is to design a PID control law that allows the output to track a reference signal  $r$  while guaranteeing stability of the closed-loop system and achieving optimal transient performance. To deal with the

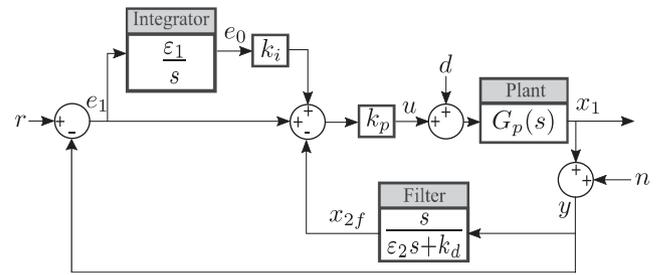


FIGURE 1 Diagram of the closed-loop system.

effect of the noise, the derivative of the measured output is filtered through a low-pass filter. The block diagram of the closed-loop system is shown in Figure 1. Accordingly, the objective is to design  $k_p$ ,  $k_i$ ,  $k_d$ ,  $\epsilon_1$ , and  $\epsilon_2$  so that  $y$  tracks a reference signal  $r$ .

Achieving an optimal transient performance requires solving a MOP for the controller parameter. This means selecting the PID controller parameters plus the filter time constant to satisfy a number of objectives including minimizing the output overshoot, peak time, while minimizing the effect of disturbance and measurement noise. As it is well-known, these objectives are in conflict with each other. This calls for employing a multi-objective optimization technique to select the four controller parameters for a four-dimensional system, which is comprised of the system dynamics (1)–(3) in addition to the integrator and filter dynamics.

## 3 | THREE TIME SCALES DESIGN

We will solve the control problem by employing the feedback control strategy shown in Figure 1. Key to this strategy is the use of the PID controller along with a filter to alleviate the effect of differentiating the noise. Following this approach, we will have five parameters to design, namely,  $k_p$ ,  $k_i$ ,  $k_d$ ,  $\epsilon_1$ , and  $\epsilon_2$ . This problem has been solved in Sardahi and Boker [19] with  $\epsilon_1 = 1$  using multi-objective optimization technique. In Sardahi and Boker [19], the problem was solved in one shot by optimizing the controller parameters to achieve a number of conflicting objectives. Depending on the application and the plant, this approach might be computationally prohibitive, especially for large-scale systems. To overcome this limitation, we will solve this problem in a three-time scale fashion by employing the singular perturbation technique [26] to design the control system. Accordingly, we propose the control law

$$u = k_p [(k_i e_0 + e_1) - x_{2f}], \quad (4)$$

where  $e_1 = r - y = r - x_1 - n$  and  $e_0$  and  $x_{2f}$  are the states of integrator and filter, respectively, as indicated in Figure 1.

We now augment the plant with the integrator and filter dynamics and write the closed-loop system as

$$\dot{e}_0 = \varepsilon_1 e_1, \quad (5)$$

$$\dot{x}_1 = x_2, \quad (6)$$

$$\dot{x}_2 = a_1 x_1 + a_2 x_2 + b [k_p (k_i e_0 + e_1 - x_{2f}) + d], \quad (7)$$

$$\varepsilon_2 \dot{x}_{2f} = -k_d x_{2f} + x_2 + \dot{n}. \quad (8)$$

Note that the upper script symbol dot denotes differentiation with respect to time scale  $t$ .

The time separation is done by first making sure the integrator is slow relative to the dynamics of the system. This implies making the rate of change of the integrator (the  $e_0$  dynamics or equation (6)) to be smaller by at least an order of magnitude than that of the  $x$  dynamics or equations (7), (8). This means making the rate of change of (6) of the order  $\mathcal{O}(\varepsilon_1)$ , where  $\varepsilon_1 < 1$ , while the order of the rate of change of the dynamics of  $x$  is  $\mathcal{O}(1)$ . Second, we design the filter (equation (9)) to be faster than the system dynamics by an order of magnitude. This means making the rate of change of (6) of the order  $\mathcal{O}(1/\varepsilon_2)$ , where  $\varepsilon_2 < 1$ . Finally, to make sure that the time scale of the whole system is well defined and that the filter is faster than the integrator, we choose  $0 < \varepsilon_2 \ll \varepsilon_1 < 1$ . Accordingly, the design procedure will be done in three time scales separately. First, we analyze the system in the slow time scale, denoted by  $\theta$  and defined by  $\theta = t\varepsilon_1$ . As a result, the system dynamics in the  $\theta$  time scale can be written as

$$\frac{de_0}{d\theta} = e_1, \quad (9)$$

$$\varepsilon_1 \frac{dx_1}{d\theta} = x_2, \quad (10)$$

$$\varepsilon_1 \frac{dx_2}{d\theta} = a_1 x_1 + a_2 x_2 + b [k_p (k_i e_0 + e_1 - x_{2f}) + d], \quad (11)$$

$$\varepsilon_1 \varepsilon_2 \frac{dx_{2f}}{d\theta} = -k_d x_{2f} + x_2 + \varepsilon_1 \frac{dn}{d\theta}. \quad (12)$$

To find the reduced (slow) subsystem, we let  $\varepsilon_1 \rightarrow 0$  in (9)-(12). This leads to

$$x_2^s = 0, \quad x_{2f}^s = \frac{x_2^s}{k_d} = 0, \quad (13)$$

$$e_1^s = \frac{a_1 r - a_1 n + b k_p k_i e_0^s + b d}{(a_1 - b k_p)}, \quad (14)$$

which in turn leads to the reduced subsystem

$$\frac{de_0^s}{d\theta} = \frac{a_1 r - a_1 n + b k_p k_i e_0^s + b d}{(a_1 - b k_p)}, \quad e_0^s(0) = e_0(0). \quad (15)$$

Note that the superscript  $s$  used in (13)-(15) emphasizes the fact that these states belong to the slow time scale. It should also be noted that (15) shows that the system in this time scale has scalar dynamics that depend on the noise and disturbance with  $k_p$  and  $k_i$  being the only design parameters.

We next analyze the system in the plant time scale  $t$ . In this case, we let  $\varepsilon_1 \rightarrow 0$ , which also implies  $\varepsilon_2 \rightarrow 0$ , in (5)-(8) so that we obtain

$$\dot{e}_0 = 0 \Rightarrow e_0 = e_0^s = \text{constant}, \quad (16)$$

$$x_{2f} = \frac{1}{k_d}(x_2 + \dot{n}). \quad (17)$$

Notice that due to the continuity of the system and as it is customary in singular perturbation analysis, we have (16) indicating that the slow integrator state is considered to be "frozen" in this time scale. Similarly, we regard  $e_1^s, x_1^s, x_2^s$  and  $x_{2f}^s$  as frozen in the  $t$  time scale, which means their derivative with respect to  $t$  is zero. In view of (17), we further have

$$\dot{x}_1 = x_2, \quad (18)$$

$$\begin{aligned} \dot{x}_2 = & (a_1 - b k_p) x_1 + \left( a_2 - \frac{b k_p}{k_d} \right) x_2 + b k_p k_i e_0^s \\ & + b k_p r - b k_p n - \frac{b k_p}{k_d} \dot{n} + b d \end{aligned} \quad (19)$$

The term dependent on  $e_0^s$  in (19) is regarded as bias since  $e_0^s$  is a 'frozen' parameter in this time scale. Consequently, to remove this bias from (18)-(19), we employ

$$x_1^f = x_1 + \frac{b k_p k_i e_0^s}{(a_1 - b k_p)}, \quad x_2^f = x_2 - x_2^s = x_2, \quad (20)$$

$$x_{2f}^f = x_{2f} - x_{2f}^s = x_{2f},$$

where the slow bias in  $x_2$  and  $x_{2f}$  is zero as derived in (13). We now use (20) to get the  $t$ -time scale reduced system

$$\dot{x}_1^f = x_2^f, \quad x_1^f(0) = x_1(0) + \frac{b k_p k_i e_0^s}{(a_1 - b k_p)} e_0^s(0), \quad (21)$$

$$\begin{aligned} \dot{x}_2^f = & (a_1 - b k_p) x_1^f + \left( a_2 - \frac{b k_p}{k_d} \right) x_2^f + b k_p r \\ & - b k_p n - \frac{b k_p}{k_d} \dot{n} + b d, \quad x_2^f(0) = x_2(0), \end{aligned} \quad (22)$$

with

$$x_{2f}^f = \frac{1}{k_d}(x_2^f + \dot{n}). \quad (23)$$

Finally, we study the system in the fastest time scale, which we define as  $\sigma = \frac{t}{\varepsilon_2}$ . The closed-loop dynamics, in this case, will be as follows:

$$\frac{de_0}{d\sigma} = \varepsilon_1 \varepsilon_2 e_1, \quad (24)$$

$$\frac{dx_1}{d\sigma} = \varepsilon_2 x_2, \quad (25)$$

$$\frac{dx_2}{d\sigma} = \varepsilon_2 \left[ a_1 x_1 + a_2 x_2 + b k_p (k_i e_0 + e_1 - x_{2f}) + b d \right], \quad (26)$$

$$\frac{dx_{2f}}{d\sigma} = -k_d x_{2f} + x_2 + \frac{1}{\varepsilon_2} \frac{dn}{d\sigma}. \quad (27)$$

To find the fastest subsystem, we define

$$x_2^{ff} = x_{2f} - x_{2f}^f, \quad (28)$$

where  $x_{2f}^f$  is defined in (17). Note that, in view of  $x_2 = x_2^f$  from (20) and in the  $\sigma$  time scale, (17) can be written as  $x_{2f}^f = \frac{1}{k_p} \left( x_2 + \frac{1}{\varepsilon_2} \frac{dn}{d\sigma} \right)$ . Using this change of variables, substitute by (28) in (27). This eliminates  $\varepsilon_2$  from (27). Now let  $\varepsilon_2 \rightarrow 0$  in (24)–(27) so that we have

$$\frac{de_0}{d\sigma} = \frac{dx_1}{d\sigma} = \frac{dx_2}{d\sigma} = 0, \quad (29)$$

and

$$\frac{dx_{2f}^{ff}}{d\sigma} = -k_d x_{2f}^{ff}, \quad x_{2f}^{ff}(0) = x_{2f}(0) - x_{2f}^f(0). \quad (30)$$

The decomposition of the three-time scale system is now summarized in the following proposition.

**Proposition 1.** *In the limit as  $\varepsilon_1 \rightarrow 0$  and  $\varepsilon_2 \rightarrow 0$  and for all  $0 < \varepsilon_2 \ll \varepsilon_1 < 1$  and for all  $t \geq 0$ , the closed-loop system (5)–(8) is decomposed into the following three subsystems:*

1. *the slow subsystem*

$$\frac{de_0^s}{d\theta} = \frac{a_1 r - a_1 n + b k_p k_i e_0^s + b d}{(a_1 - b k_p)}, \quad e_0^s(0) = e_0(0). \quad (31)$$

2. *the fast subsystem*

$$\begin{aligned} \dot{x}_1^f &= x_2^f, \\ \dot{x}_2^f &= (a_1 - b k_p) x_1^f + \left( a_2 - \frac{b k_p}{k_d} \right) x_2^f + b k_p r \\ &\quad - b k_p n - \frac{b k_p}{k_d} \dot{n} + b d, \end{aligned} \quad (32)$$

$$x_1^f(0) = x_1(0) + \frac{b k_p k_i e_0^s}{(a_1 - b k_p)}, \quad x_2^f(0) = x_2(0).$$

3. *the fastest subsystem*

$$\frac{dx_{2f}^{ff}}{d\sigma} = -k_d x_{2f}^{ff}, \quad x_{2f}^{ff}(0) = x_{2f}(0) - x_{2f}^f(0). \quad (33)$$

*Remark 1.* Singular perturbation theory [26] suggests that the responses of the reduced subsystems (31)–(33) become close to the response of the closed-loop system (5)–(8) as  $\varepsilon_2$  and  $\varepsilon_1$  become sufficiently small. This is shown in the case of  $\varepsilon_2 = 1$ , and the measurement noise  $n$  and its derivative are both zero-mean stationary white noise [Theorem 2.1, Ch. 4 [26]]. In this case, the closed-loop system exhibits a two-time scale structure. This result further implies that if the parameters  $k_p, k_d, k_i$  are chosen such that the reduced subsystems are asymptotically stable, then there exists an  $\varepsilon_1^*$  such that for all  $\varepsilon_1 \in (0, \varepsilon_1^*)$  and for all  $t \geq 0$ , the error between the original states of systems (5)–(8) and the states of the reduced subsystems to be  $O(\varepsilon_1^{1/2})$ . This means that as  $\varepsilon_1$  gets smaller, the performance of the reduced subsystems gets close to the original one.

*Remark 2.* It is clear from Proposition 1 that the closed-loop systems (5)–(8) are decomposed into three lower dimensional and decoupled systems. This allows for the design of the the controller parameters in the following way:

1. Consider subsystem (32) and design  $k_p$  and  $k_d$  by solving a MOP to optimize a number of conflicting objectives including the minimization of the effect of noise and disturbance. The choice of  $k_p$  and  $k_d$  needs to be constrained to guarantee stability of (32). In addition,  $k_d$  needs to be chosen to be positive to guarantee stability of (33).
2. Step 1 results in optimal values for  $k_d$  and  $k_p$ . We then use the values obtained for  $k_p$  to design  $k_i$  so to guarantee stability of (31).
3. Choose  $[\varepsilon_1, \varepsilon_2]$  small enough for acceptable performance.

This procedure is illustrated by a simulation example in Section 5.

*Remark 3.* It is important to note that the change of variables (20) does not depend on the disturbance nor noise. This allows for design and analysis to be performed on the reduced system (32) and then similar performance is expected for the original system for sufficiently small  $\varepsilon_1$ . This is different than the change of variables suggested by Kokotović et al. [26] to analyze the two-time scale PID controller in the absence of measurement noise and filter. In that case, the

objective was to show that the disturbance effect can be decoupled in different time scales.

## 4 | MULTI-OBJECTIVE OPTIMIZATION

### 4.1 | Basic concept

Multi-objective optimization is an area of multiple criteria decision-making. It deals with mathematical optimization problems involving two or more conflicting fitness functions to be optimized at the same time. MOPs have gained a lot of attention recently because of their enormous applications in engineering, economics and logistics, just to name a few. In mathematical terms, a MOP can be stated as follows:

$$\min_{\mathbf{k} \in Q} \{\mathbf{F}(\mathbf{k})\}, \quad (34)$$

where  $\mathbf{F} : Q \rightarrow \mathbf{R}^k$ ,  $\mathbf{F}(\mathbf{k}) = [f_1(\mathbf{k}), \dots, f_k(\mathbf{k})]$  is the map that consists of the objective functions  $f_i : Q \rightarrow \mathbf{R}^1$  under consideration and  $\mathbf{k} \in Q$  is a  $q$ -dimensional vector of design parameters. The domain  $Q \subset \mathbf{R}^q$  can in general be expressed by inequality and equality constraints:

$$Q = \{\mathbf{k} \in \mathbf{R}^q \mid g_i(\mathbf{k}) \leq 0, i = 1, \dots, l, \text{ and } h_j(\mathbf{k}) = 0, j = 1, \dots, m\}. \quad (35)$$

The solution of MOPs comprises a set known as the *Pareto set* and the corresponding set of objective values called *Pareto front*. The optimal solution is obtained using the *dominancy* concept [27]. For a comprehensive survey of the methods used to solve the MOPs problems, the reader can consult [28–30]. In this paper, the NSGA-II is used to solve the MOP at hand. NSGA-II is the improved version of NSGA, which is faster, has a better sorting algorithm, includes an elitism mechanism, and eliminates the need for the sharing parameter that was introduced in NSGA [25]. We next provide more details about the algorithm.

### 4.2 | NSGA-II

The NSGA-II algorithm consists of eight basic operations: initialization, fitness evaluation, domination ranking, crowding distance calculation, selection, crossover, mutation, and combination to solve MOPs [25] as depicted in Figure 2. The initialization algorithm is executed first to generate a random population matrix,  $\mathbf{Pop}$ , within the specified lower and upper bounds of the tuning parameters. This matrix is passed to the fitness algorithm,  $\mathbf{F}(\mathbf{Pop})$ , to compute the cost functions at each candidate solution. Then, the ranking algorithm is called to place the candidate solutions at different fronts based on their

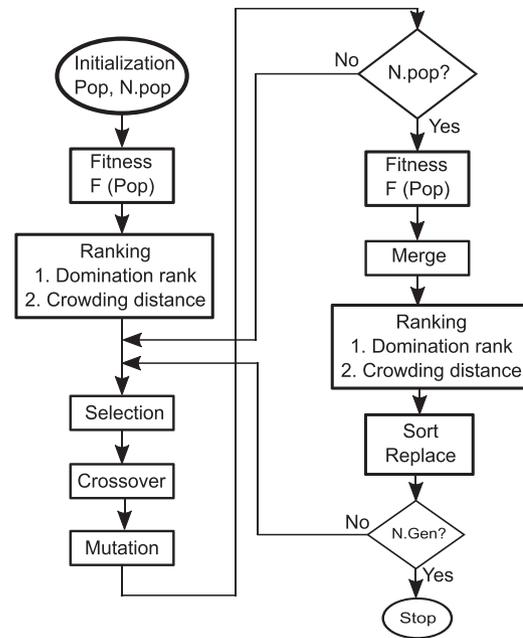


FIGURE 2 Flow chart of nondominated sorting genetic algorithm (NSGA-II).

nondomination ranks and crowding distance values. That is, the solutions in the first front dominate all the other solutions, while the solutions in the second front are dominated only by the members in the first front, and so on. To ensure a diverse and distributed solution, the crowding distance is calculated for each solution front-wise. The crowding distance measures how far a solution is from its neighboring solutions within the same front. Larger crowding distance values indicate a better distribution of the solutions. After ranking, the selection algorithm is called to randomly select parents that will be used to produce the next generation of candidate solutions. To this end, the binary tournament selection algorithm is usually used. The selection process starts with generating two uniformly distributed random integer numbers between 1 and  $N.pop$  (number of candidate solutions). The two numbers are used to select two candidate parents from  $\mathbf{Pop}$ . The parent with either a smaller rank or bigger crowding distance than the other is chosen. The selection process continues for  $N.pop$  times. After that, the crossover process such as the arithmetic crossover algorithm [31, 32] and a mutation algorithm such as the simple mutation approach [33] are executed using the selected parents to produce children. These two operations are executed  $N.pop/2$  times producing a new offspring of size  $N.pop$ . After that, the children are merged with the current population. This combination guarantees the elitism of the best individuals. Finally, the merged population is sorted based on crowding distance and rank values. The new generation is produced from the sorted population until the number of population reaches  $N.pop$ .

The selection, crossover, mutation, merging, ranking, and sorting process are repeated over and over until the number of generations reaches a preselected  $N.Gen$ .

### 4.3 | Multi-objective optimal control

The closed-loop system should be designed such that the effect of  $D(s)$  and  $N(s)$  on the output is minimal. Here,  $D(s)$  and  $N(s)$  are the Laplace transforms of  $n(t)$  and  $d(t)$ , respectively, shown in Figure 1. However, these two objectives are conflicting and cannot be made minimum at the same time. Instead, optimal trade-off solutions between them can be found. Another example of conflicting design goals is the problem of maximizing the controlled system speed of response (by minimizing the peak time  $t_p$ ) and minimizing the overshoot ( $M_p$ ). Thus, a MOP should be formulated to handle these conflicting design objectives. To account for the impact of  $N(s)$  and  $D(s)$  on the performance of the closed-loop system and numerically compute  $t_p$  and  $M_p$ , we define with the help of Figure 1 three functions: the complementary sensitivity transfer function  $CSTF$ , the load disturbance sensitivity transfer function  $LDSTF$ , and the noise sensitivity transfer function  $NSTF$  as follows:

$$CSTF(s) = \frac{C(s)G_p(s)}{1 + C(s)G_p(s)F(s)}, \quad (36)$$

$$LDSTF(s) = \frac{G_p(s)}{1 + C(s)G_p(s)F(s)}, \quad (37)$$

$$NSTF(s) = \frac{C(s)G_p(s)F(s)}{1 + C(s)G_p(s)F(s)}, \quad (38)$$

where  $C(s) = k_p \left(1 + \varepsilon_1 \frac{k_i}{s}\right)$ ,  $G_p(s) = \frac{b}{s^2 - a_2 s - a_1}$  and  $F(s) = 1 + \left(\frac{s}{\varepsilon_2 s + k_d}\right) \left(\frac{s}{s + \varepsilon_1 k_i}\right)$ .

The transfer functions (36)–(38) are used for the controller design in the following manner; first, the step response of (36) is simulated and used to compute  $t_p$  and  $M_p$ , which can then be minimized. Similarly, the frequency responses of (38) and (37) are generated and used to minimize the impact of  $N(s)$  and  $D(s)$  on the closed-loop control system. This procedure is illustrated further in the Section 5.

Our strategy is to use the three lower dimensional subsystems (31)–(33) to solve the multi-objective design problem and then we select  $\varepsilon_1$  and  $\varepsilon_2$  small enough to recover the desired performance. Towards this goal, we note that the fast subsystem has only two design parameters ( $k_p$  and  $k_d$ ), which is very attractive from the optimization perspective since it leads to less computation load. Using this subsystem, the complementary sensitivity transfer function  $CSTF^f$ , the load disturbance sensitivity transfer function  $LDSTF^f$ , and the noise sensitivity

transfer function  $NSTF^f$  of the fast subsystem read

$$CSTF^f(s) = \frac{X_1^f(s)}{R(s)} = -\frac{bk_p}{s^2 - \left(a_2 - \frac{bk_p}{k_d}\right)s - (a_1 - bk_p)}, \quad (39)$$

$$LDSTF^f(s) = \frac{X_1^f(s)}{D(s)} = \frac{b}{s^2 - \left(a_2 - \frac{bk_p}{k_d}\right)s - (a_1 - bk_p)}, \quad (40)$$

$$NSTF^f(s) = \frac{X_1^f(s)}{N(s)} = -\frac{bk_p \left(1 + \frac{s}{k_d}\right)}{s^2 - \left(a_2 - \frac{bk_p}{k_d}\right)s - (a_1 - bk_p)}. \quad (41)$$

It is worth noting that the three functions (39)–(41) can also be obtained from Figure 1 by setting  $\varepsilon_1 = \varepsilon_2 = 0$  and simplifying the block diagram. The functions (39)–(41) can now be used to solve the MOP.

## 5 | SIMULATION EXAMPLE: DC SERVO MOTOR

### 5.1 | System models

Consider a DC servo motor system represented by the equations

$$\dot{x}_1 = x_2, \quad (42)$$

$$\dot{x}_2 = -\frac{1}{\tau}x_2 + \frac{K}{\tau}(u + d), \quad (43)$$

$$y = x_1 + n, \quad (44)$$

where  $x = [x_1 \ x_2]^T$  is the vector of the system states,  $u$  is the control input,  $y$  is the measured output,  $n$  is measurement noise, and  $d$  is the unknown external disturbances, that is, unknown load torque. It is assumed that  $d$  and its derivatives are bounded.

The control law defined in (4) will be used to track the reference signal  $r$ . The parameters  $K_p$ ,  $K_i$ ,  $K_d$ ,  $\varepsilon_1$ , and  $\varepsilon_2$  will be designed according to the procedure defined in Remark 2 in Section 3 and following the multi-objective optimization approach described in Section 4.

By comparing the DC motor model in (42)–(44) with the generic model in (1)–(3), we have the following correspondence;  $a_1 = 0$ ,  $a_2 = -\frac{1}{\tau}$ ,  $b = \frac{K}{\tau}$ . Using these values and in view of the singular perturbation decomposition (31)–(33), the slow subsystem is given by

$$\frac{de_0^s}{d\theta} = -k_i e_0^s - \frac{1}{k_p} d, \quad (45)$$

where  $e_0^s(0) = e_0(0)$ . Likewise, the fast subsystem reads

$$\dot{x}_1^f = x_2^f, \quad (46)$$

$$\begin{aligned} \dot{x}_2^f = & -\frac{K}{\tau}k_p x_1^f - \frac{1}{\tau} \left(1 + \frac{k_p}{k_d}\right) x_2^f - \frac{K}{\tau}k_p n + \frac{K}{\tau}d \\ & - \frac{K}{\tau} \frac{k_p}{k_d} \dot{n} + \frac{K}{\tau}k_p r. \end{aligned} \quad (47)$$

$$x_1^f(0) = x_1(0) - k_i e_0^s(0), \quad x_2^f(0) = x_2(0).$$

Finally, the fastest subsystem is provided by

$$\frac{dx_{2f}^{ff}}{d\sigma} = -k_d x_{2f}^{ff}. \quad (48)$$

After obtaining the optimal solutions for  $k_p$  and  $k_d$ , we will use them to design the remaining parameters by using the slow (45) and the fastest subsystems (48). This will be explained further in the next subsection.

## 5.2 | Multi-objective optimal control design

Consider a multi-objective control design for the fast subsystem (46)–(47), where the gain  $\mathbf{k} = [k_p, k_d]$  are the design parameters. The chosen design space for the parameters is as follows,

$$Q = \{\mathbf{k} \in [1,60] \times [49,240] \subset \mathbf{R}^2\}. \quad (49)$$

The values of  $k_p$  and  $k_d$  are chosen such the stability of (46)–(47) is guaranteed. In addition,  $k_d$  is chosen to be positive to guarantee stability of (48). The upper boundaries are arbitrarily chosen. These constraints on the parameter space are used during the simultaneous minimization of the following design objectives:

$$\min_{\mathbf{k} \in Q} \left\{ t_p, M_p, \left\| LDSTF^f(j\omega) \right\|_{\infty}, \left\| NSTF^f(j\omega) \right\|_{\infty} \right\}, \quad (50)$$

where  $t_p$  is the peak time,  $M_p$  is the maximum percentage overshoot, and  $\left\| LDSTF^f(j\omega) \right\|_{\infty}$  and  $\left\| NSTF^f(j\omega) \right\|_{\infty}$  are the load disturbance and noise rejection capabilities of the closed-loop system, respectively. The terms  $\left\| LDSTF^f(j\omega) \right\|_{\infty}$  and  $\left\| NSTF^f(j\omega) \right\|_{\infty}$  are given by

$$\left\| LDSTF^f(j\omega) \right\|_{\infty} = \sup_{\omega < \omega_c} \sigma(LDSTF^f(j\omega)), \quad (51)$$

$$\left\| NSTF^f(j\omega) \right\|_{\infty} = \sup_{\omega > \omega_c} \sigma(NSTF^f(j\omega)), \quad (52)$$

where  $\sigma$  and  $\omega_c$  are the largest singular value and the threshold frequency of the transfer function, respectively. Typically, measurement noises are most influential at high frequencies, and load disturbances are most influential

at low frequencies. Accordingly, for this application, we use  $\omega_c$  to be at 1 rad/s. A controlled system is considered to have good load disturbance and noise rejection capabilities if it conforms to  $\left\| LDSTF^f(j\omega) \right\|_{\infty} \ll 1$  and  $\left\| NSTF^f(j\omega) \right\|_{\infty} \ll 1$ , respectively. Therefore, the constraints  $\left\| LDSTF^f(j\omega) \right\|_{\infty} < 1$  and  $\left\| NSTF^f(j\omega) \right\|_{\infty} < 1$  are also imposed to ensure these capabilities.

The NSGA-II is utilized to solve this multi-optimization problem. The reader can consult Deb [34] for in-depth understanding about this algorithm. According to Matlab documentation, the population size can be configured in multiple ways and by default; population size is 15 times the number of the design variables  $nvars$ . And the maximum number of generations should not exceed  $200 \times nvars$ . In this study, the population size and number of generation are set to 200.

The Matlab version of the NSGA-II is used to solve the MOP defined in (50) by tuning the parameters defined in (49). According to the literature, NSAG-II performs well on two-objective and three-objective problems. For more than three objectives, a large number of populations should be used to enhance the searchability of the algorithm [35]. Also, the algorithm converges to the true Pareto front if the number of design parameters is less than or equal to 128 [36]. In this paper, the size of the objective space is 4 and that of decision variable space is 2. The population size and number of generation are set to 200. Therefore, NSGA-II is expected to perform well at solving the optimization problem at hand.

## 5.3 | Simulation results

The Pareto fronts from the fast subsystem and the impact of  $\varepsilon_1$  and  $\varepsilon_2$  on the optimal solution and closed-loop system response at different solutions are discussed here. In the simulation results, we use  $\varepsilon_1 = 10\varepsilon_2$ . However, other choices, such that  $\varepsilon_2$  is sufficiently smaller than  $\varepsilon_1$ , will show a similar trend.

Two projections from the 4-D Pareto front are shown in Figure 3a,b. Figure 3a shows the conflicting relationship between the overshoot  $M_p$  and peak time  $t_p$ , while Figure 3b shows the conflict between the objective of minimizing the impact of measurement noise and load disturbance. The figures also show how  $\varepsilon_1$  and  $\varepsilon_2$  affect the optimal solution. For example, the lower optimal values of  $M_p$  increase as  $\varepsilon_1$  and  $\varepsilon_2$  increase. On the other side, the upper values of  $t_p$  decrease as  $\varepsilon_1$  and  $\varepsilon_2$  go up. So, a control system designer should expect smaller  $t_p$  but larger overshoot for every point on the front after adding the integral action and filter to the closed-loop response. On the other hand, Figure 3b shows that the optimal solutions from the fast subsystem represent the maximum values of  $\left\| LDSTF^f(j\omega) \right\|_{\infty}$  and  $\left\| NSTF^f(j\omega) \right\|_{\infty}$ . That is,

implementing the integral control and filter will lead to smaller values of these design objectives and in turn better rejection of external noise and disturbance.

Three points of interest are labeled in Figure 3a,b.  $P_I$  has the coordinate of the minimum objective functions and is thus an ideal point not on the Pareto front.  $P_N$ , knee point, is a point on the Pareto front and the closest to  $P_I$ . This point can be more attractive to the designer than other optimal options since it is the closest to the ideal solution.  $P_F$  is a point on the Pareto front that has the largest Euclidean distance from  $P_I$ . Even though  $P_F$  is one of the optimal solutions, it could be less attractive to the decision-maker because it is far away from the ideal solution. These points illustrate the properties of the multi-objective optimization design.

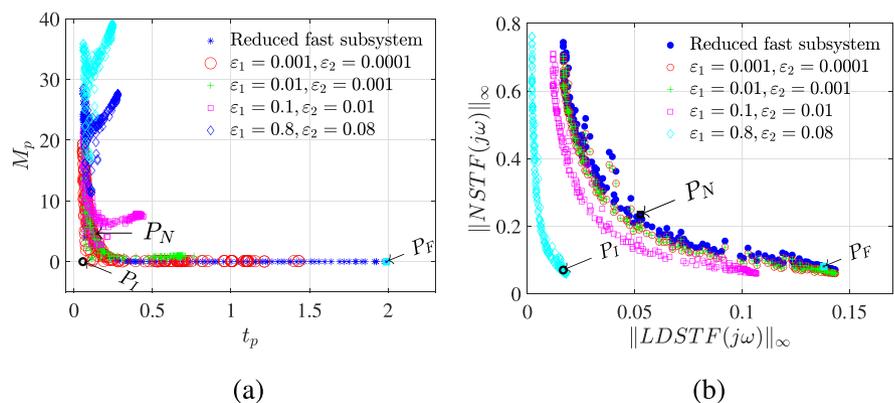
Furthermore, we notice that the results agree with the singular perturbation theory [26] that suggests that the responses of the reduced subsystems (46) and (47) become close to the response of the closed-loop system (42) and (43) as  $\varepsilon_2$  and  $\varepsilon_1$  become sufficiently small. That is, the solution of the optimization problem (50) using the fast subsystem (46) and (47) becomes closer to that of the closed-loop system (42) and (43) as the singular perturbation parameters approach zero.

Figure 4a shows the closed-loop response when  $t_p$  from the Pareto front is the largest (Figure 4a(i)) and when it is the smallest (Figure 4a(ii)). It is evident that fast system responses are associated with large overshoot values, while those at  $\max(t_p)$  are associated with smaller overshoot values. These observations agree with the results depicted in Figure 3a. Also, we notice that the overshoot goes up and peak time goes down as we increase the values of  $\varepsilon_1$  and  $\varepsilon_2$ .

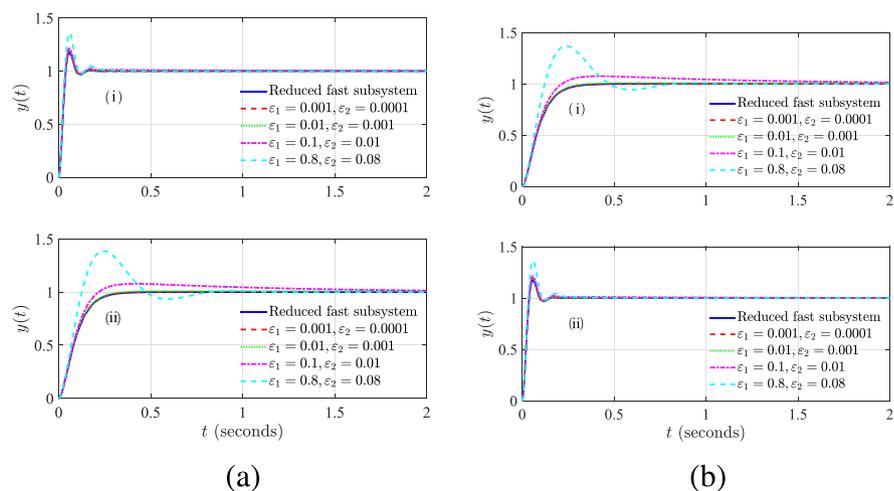
Figure 4b shows the controlled system response at the maximum and minimum values of  $M_p$ . Since there is a conflict relationship between  $M_p$  and  $t_p$ , we notice that the closed-loop behavior at  $\min(M_p)$  (Figure 4b(i)) is exactly the same as that when  $t_p$  is the largest (Figure 4a(ii)) and vice versa.

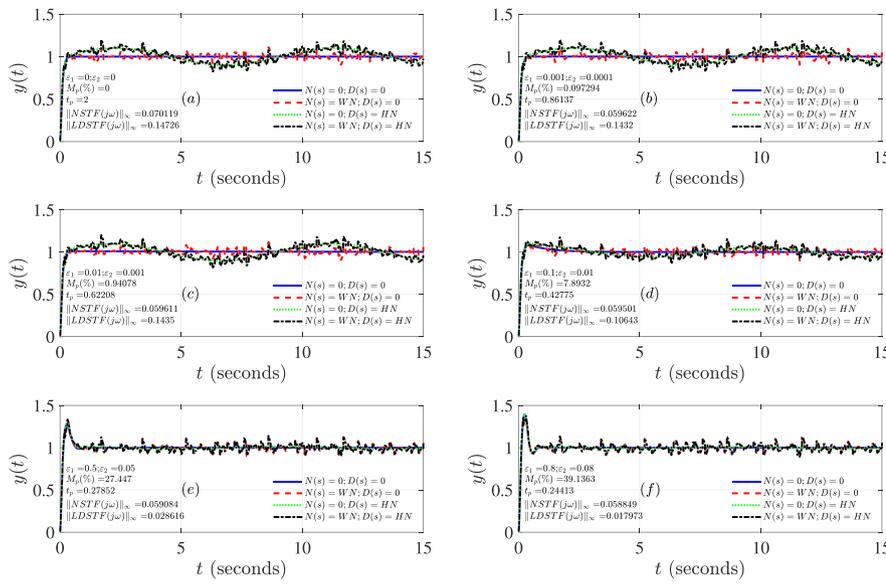
Figure 5 shows the closed-loop response at  $\min(\|NSTF^j(j\omega)\|_\infty)$  with  $k_p = 6.9854$ ,  $k_d = 236.6588$  and  $k_i = 10$  for different values of  $\varepsilon_1$  and  $\varepsilon_2$ . Here, the noise signal is assumed to be a white noise with variance 0.0001 and zero mean. As evident from Figure 5a, the response at  $\min(\|NSTF^j(j\omega)\|_\infty)$  is associated with an increase in the percentage overshoot  $M_p$  but smaller  $t_p$  and better rejection of noise and disturbance as the singular perturbation parameters get large. Figure 6, on the other hand, shows the closed-loop response for  $\min(\|LDSTF^j(j\omega)\|_\infty)$

**FIGURE 3** (a)  $M_p$  versus  $t_p$  for different values of  $\varepsilon_1$  and  $\varepsilon_2$ . (b)  $\|NSTF^j(j\omega)\|_\infty$  versus  $\|LDSTF^j(j\omega)\|_\infty$  for different values of  $\varepsilon_1$  and  $\varepsilon_2$ . Here  $P_I$  is the ideal solution,  $P_N$  is the knee point, and  $P_F$  is the far-point. [Color figure can be viewed at [wileyonlinelibrary.com](http://wileyonlinelibrary.com)]

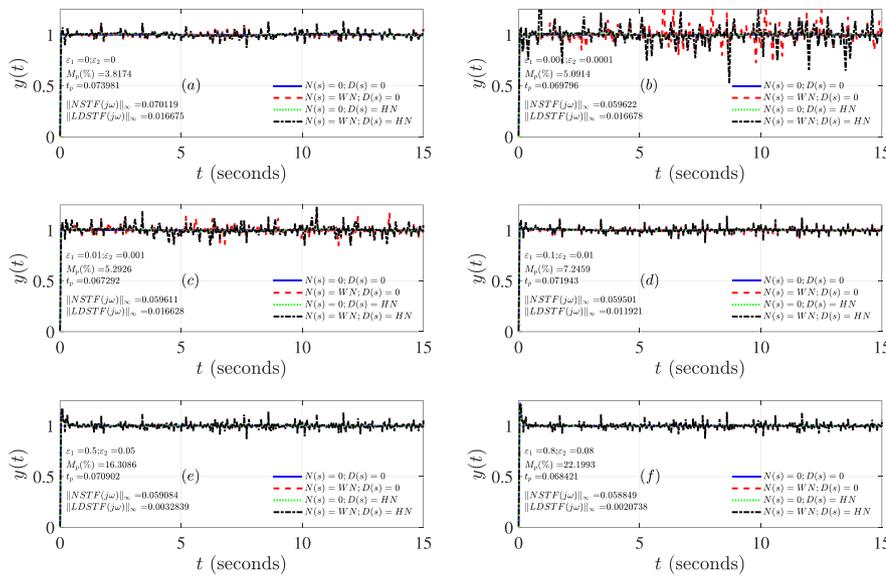


**FIGURE 4** (a) Closed-loop system response at (i)  $\min(t_p)$  with  $k_p = 59.9394$ ,  $k_d = 238.2716$  and (ii)  $\max(t_p)$  with  $k_p = 7.2439$  and  $k_d = 129.2492$ . (b) Closed-loop system response at (i)  $\min(M_p)$  with  $k_p = 8.1148$ ,  $k_d = 68.8419$  and (ii)  $\max(M_p)$  with  $k_p = 29.9168$ ,  $k_d = 238.7562$ . [Color figure can be viewed at [wileyonlinelibrary.com](http://wileyonlinelibrary.com)]

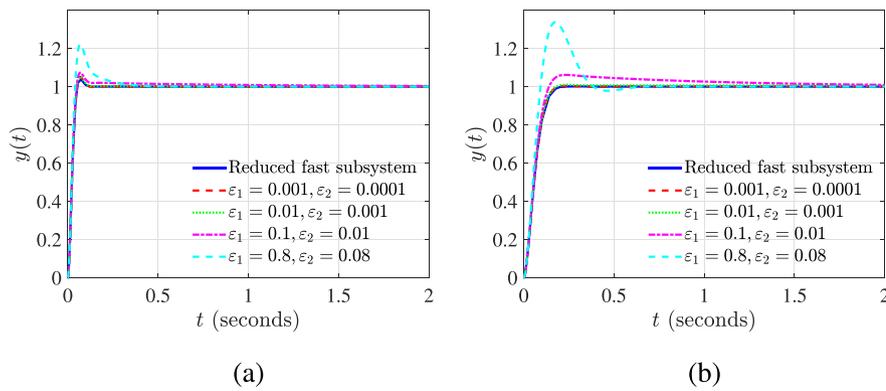




**FIGURE 5** Closed-loop system response at  $\min(\|NSTF^j(j\omega)\|_\infty)$  with  $k_p = 6.9854$ ,  $k_d = 236.6588$  and  $k_i = 10$  (a) with  $\epsilon_1 = 0$ ,  $\epsilon_2 = 0$ ; (b) with  $\epsilon_1 = 0.001$ ,  $\epsilon_2 = 0.0001$ ; (c) with  $\epsilon_1 = 0.01$ ,  $\epsilon_2 = 0.001$ ; (d) with  $\epsilon_1 = 0.1$ ,  $\epsilon_2 = 0.01$ ; (e) with  $\epsilon_1 = 0.5$ ,  $\epsilon_2 = 0.05$ ; and (f) with  $\epsilon_1 = 0.8$ ,  $\epsilon_2 = 0.08$ . [Color figure can be viewed at wileyonlinelibrary.com]



**FIGURE 6** Closed-loop system response at  $\min(\|LDSTF^j(j\omega)\|_\infty)$  with  $k_p = 59.9720$ ,  $k_d = 92.4594$  and  $k_i = 10$  (a) with  $\epsilon_1 = 0$ ,  $\epsilon_2 = 0$ ; (b) with  $\epsilon_1 = 0.001$ ,  $\epsilon_2 = 0.0001$ ; (c) with  $\epsilon_1 = 0.01$ ,  $\epsilon_2 = 0.001$ ; (d) with  $\epsilon_1 = 0.1$ ,  $\epsilon_2 = 0.01$ ; (e) with  $\epsilon_1 = 0.5$ ,  $\epsilon_2 = 0.05$ ; and (f) with  $\epsilon_1 = 0.8$ ,  $\epsilon_2 = 0.08$ . [Color figure can be viewed at wileyonlinelibrary.com]



**FIGURE 7** (a) Closed-loop system response at  $P_N$  with  $k_p = 59.9720$  and  $k_d = 92.4594$ . (b) Closed-loop system response at  $P_F$  with  $k_p = 12.2549$  and  $k_d = 102.2563$ . [Color figure can be viewed at wileyonlinelibrary.com]

with  $k_p = 59.9720$ ,  $k_d = 92.4594$  and  $k_i = 10$ . In this case, the load disturbance is modeled by a harmonic signal,  $d(s) = 0.7 \times \sin(0.7t)$  represented by HN in figures. As

the singular perturbation parameters get large, we notice better disturbance rejection, but at the expense of  $M_p$ . On other side,  $t_p$  gets smaller which indicates faster response.

This is expected since larger  $\varepsilon_1$  indicates larger integration action, which is known to make the overshoot goes up. As a result, the peak time goes down.

Figure 7a shows the closed-loop response at  $P_N$ , while Figure 7b depicts the controlled system response at  $P_F$ . In both figures, we notice that the closed-loop response shows faster response and larger overshoot as  $\varepsilon_1$  and  $\varepsilon_2$  increase. However, the optimal solution at  $P_N$  shows less increase in  $M_p$  as the integral and filter terms enter the system.

Moreover, it should be pointed out that the proposed three-time scale design reduces the computation time significantly. On Core-i7 PC with four cores, eight logical processors, and 1.99 GHz base speed, the solution of the MOP under three-time scale separation takes about 2 min, while it increases to 34 min when all design parameters ( $k_p$ ,  $k_d$ ,  $k_i$ , and  $\varepsilon_2$ ) are considered at the same time, as reported in Sardahi and Boker [19].

## 6 | CONCLUSION

We have studied the multi-objective optimal design of the PID controller in three time scales for a system having a load disturbance and corrupted feedback signal. An optimization problem with four design objectives and two design parameters is solved by the NSGA-II algorithm. The results show that the optimal solutions obtained by using the reduced fast subsystem lead to better rejection of noise and load disturbance and faster closed-loop response but higher overshoot values than those on the front after adding the integral action and filter. Also, the results show that the Pareto front of the closed-loop system gets closer to that of the reduced subsystem as the singular perturbation parameters become sufficiently small. For future work, it is interesting to investigate how the proposed scheme extends to the case of nonlinear systems and systems with more than second-order dynamics.

### AUTHOR CONTRIBUTIONS

**Muhammad Aafaque:** Formal analysis; investigation; project administration; resources; software; writing—original draft; writing—review and editing. **Almuatazbellah Boker:** Conceptualization; formal analysis; investigation; methodology; project administration; resources; supervision; writing—original draft; writing—review and editing. **Yousef Sardahi:** Formal analysis; investigation; methodology; resources; software; supervision; validation; visualization; writing—original draft; writing—review and editing.

### CONFLICT OF INTEREST STATEMENT

The authors declare no potential conflict of interests.

### ORCID

Muhammad Aafaque  <https://orcid.org/0000-0003-0439-1646>

### REFERENCES

1. R. P. Borase, D. K. Maghade, S. Y. Sondkar, and S. N. Pawar, *A review of PID control, tuning methods and applications*, Int. J. Dyn. Control **9** (2021), no. 2, 818–827.
2. N. H. Sahrir and M. A. Mohd Basri, *Modelling and manual tuning PID control of quadcopter*, *Control, instrumentation and mechatronics: Theory and practice*, Springer, 2022, pp. 346–357.
3. N. P. Lawrence, M. G. Forbes, P. D. Loewen, D. G. McClement, J. U. Backström, and R. B. Gopaluni, *Deep reinforcement learning with shallow controllers: An experimental application to PID tuning*, Control Eng. Pract. **121** (2022), 105046.
4. W. Zhang, Y. Cui, and X. Ding, *An improved analytical tuning rule of a robust PID controller for integrating systems with time delay based on the multiple dominant pole-placement method*, Symmetry **12** (2020), no. 9, 1449.
5. N. Zhu, X.-T. Gao, and C.-Q. Huang, *A data-driven approach for on-line auto-tuning of minimum variance PID controller*, ISA Trans. **130** (2022), 325–342.
6. J. G. Ziegler, N. B. Nichols, et al., *Optimum settings for automatic controllers*, Trans. ASME **64** (1942), no. 11, 759–768.
7. S. Bucz and A. Kozáková, *Advanced methods of PID controller tuning for specified performance*, PID Control Industr. Processes (2018), 73–119.
8. M. Huba, S. Chamraz, P. Bistak, and D. Vrancic, *Making the PI and PID controller tuning inspired by Ziegler and Nichols precise and reliable*, Sensors **21** (2021), no. 18, 6157.
9. B. Verma and P. K. Padhy, *Robust fine tuning of optimal PID controller with guaranteed robustness*, IEEE Trans. Industr. Electron. **67** (2019), no. 6, 4911–4920.
10. H. R. Ossareh, S. Wisotzki, J. B. Seeds, and M. Jankovic, *An internal model control-based approach for characterization and controller tuning of turbocharged gasoline engines*, IEEE Trans. Control Syst. Technol. **29** (2019), no. 2, 866–875.
11. R. Matuš, B. Šenol, B. B. Alagoz, and L. Pekař, *Design of robust PI controllers for interval plants with worst-case gain and phase margin specifications in presence of multiple crossover frequencies*, IEEE Access **10** (2022), 67713–67726.
12. C. P. B. Barros, P. R. Barros, and J. S. da Rocha Neto, *Loop shaping for PID controller design based on time and frequency specifications*, IFAC-PapersOnLine **51** (2018), no. 4, 592–597.
13. Y. Feng, X. Chen, and G. Gu, *Multiobjective  $H_2 / H_\infty$  control design subject to multiplicative input dependent noises*, SIAM J. Control Optim. **56** (2018), no. 1, 253–271.
14. S. Sui, C. L. P. Chen, and S. Tong, *Fuzzy adaptive finite-time control design for nontriangular stochastic nonlinear systems*, IEEE Trans. Fuzzy Syst. **27** (2018), no. 1, 172–184.
15. K. Li, Y. Zhang, S. Wei, and H. Yue, *Evolutionary algorithm-based friction feedforward compensation for a pneumatic rotary actuator servo system*, Appl. Sci. **8** (2018), no. 9, 1623.
16. J. Yoon and J. Doh, *Optimal PID control for hovering stabilization of quadcopter using long short term memory*, Adv. Eng. Inform. **53** (2022), 101679.
17. V. R. Segovia, T. Häggglund, and K. J. Åström, *Measurement noise filtering for PID controllers*, J. Process Control **24** (2014), no. 4, 299–313.

18. I. Tejado, B. M. Vinagre, J. E. Traver, J. Prieto-Arranz, and C. Nuevo-Gallardo, *Back to basics: Meaning of the parameters of fractional order PID controllers*, *Mathematics* **7** (2019), no. 6, 530.
19. Y. Sardahi and A. Boker, Multi-objective optimal design of four-parameter pid controls, *Dynamic systems and control conference*, Vol. **51890**, American Society of Mechanical Engineers, 2018, pp. V001T01A001.
20. M. A. Siddiqui, M. N. Anwar, and S. H. Laskar, *Tuning of PIDF controller in parallel control structure for integrating process with time delay and inverse response characteristic*, *J. Control Autom. Electr. Syst.* **31** (2020), no. 4, 829–841.
21. Y. Mchaouar, A. Abouloifa, I. Lachkar, M. Fettach, F. Giri, C. Taghzaoui, and A. Elallali, *Nonlinear control of single-phase shunt active power filters with theoretical analysis via singular perturbation*, 2018 5th International Conference on Electrical and Electronic Engineering (ICEEE). IEEE, 2018, pp. 67–72.
22. P. Lorenzetti and G. Weiss, *Saturating pi control of stable nonlinear systems using singular perturbations*, *IEEE Trans. Autom. Control* **68** (2022), 867–882.
23. V. D. Yurkevich, *MIMO tracking PI/PID controller design for nonlinear systems based on singular perturbation technique*, *IFAC-PapersOnLine* **48** (2015), no. 11, 760–765.
24. Y. Pan, X. Li, and H. Yu, *Efficient PID tracking control of robotic manipulators driven by compliant actuators*, *IEEE Trans. Control Syst. Technol.* **27** (2018), no. 2, 915–922.
25. K. Deb, A. Pratap, S. Agarwal, and T. A. M. T. Meyarivan, *A fast and elitist multiobjective genetic algorithm: NSGA-II*, *IEEE Trans. Evol. Comput.* **6** (2002), no. 2, 182–197.
26. P. Kokotović, H. K. Khalil, and J. O'reilly, *Singular perturbation methods in control: Analysis and design*, SIAM, 1999.
27. V. Pareto, *Manual of political economy*, The MacMillan Press, London, 1971. (original edition in French in 1927).
28. Z. N. Ansari and S. D. Daxini, *A state-of-the-art review on meta-heuristics application in remanufacturing*, *Arch. Comput. Methods Eng.* **29** (2022), no. 1, 427–470.
29. Z.-C. Qin, Y. Xin, and J.-Q. Sun, *Multi-objective optimal motion control of a laboratory helicopter based on parallel simple cell mapping method*, *Asian J. Control* **22** (2020), no. 4, 1565–1578.
30. N. Saini and S. Saha, *Multi-objective optimization techniques: A survey of the state-of-the-art and applications*, *Eur. Phys. J. Special Top.* **230** (2021), no. 10, 2319–2335.
31. H.-G. Beyer and K. Deb, *On self-adaptive features in real-parameter evolutionary algorithms*, *IEEE Trans. Evol. Comput.* **5** (2001), no. 3, 250–270.
32. K. Deb and R. B. Agrawal, *Simulated binary crossover for continuous search space*, *Complex Syst.* **9** (1995), no. 2, 115–148.
33. M. R. O. G. Kakde, *Survey on multiobjective evolutionary and real coded genetic algorithms*, *Proceedings of the 8th Asia Pacific Symposium on Intelligent and Evolutionary Systems*. Citeseer, 2004, pp. 150–161.
34. K. Deb, *Multi-objective optimization using evolutionary algorithms*, Wiley, New York, 2001.
35. H. Ishibuchi, Y. Sakane, N. Tsukamoto, and Y. Nojima, *Evolutionary many-objective optimization by NSGA-II and MOEA/D with large populations*, 2009 IEEE International Conference on Systems, Man and Cybernetics. IEEE, 2009, pp. 1758–1763.
36. J. J. Durillo, A. J. Nebro, C. A. C. Coello, J. García-Nieto, F. Luna, and E. Alba, *A study of multiobjective metaheuristics when solving parameter scalable problems*, *IEEE Trans. Evol. Comput.* **14** (2010), no. 4, 618–635.

## AUTHOR BIOGRAPHIES



**Muhammad Aafaque** is an Embedded Software Engineer at the R & D center of Karl DUNGS GmbH & Co. KG., Urbach, Germany, where since 2017 he is involved in research and development of state-of-the-art industrial and commercial combustion control systems. He received his B.Eng in Industrial Electronics from N.E.D. University of Engineering & Technology, Karachi, Pakistan, in 2009, M.S. in Control Systems Engineering from National University of Sciences and Technology (NUST), Islamabad, Pakistan, in 2014, and M.Sc. in Communications Engineering from Technical University of Munich (TUM), Munich, Germany, in 2017. Mr. Aafaque had been involved in research and development of attitude and orbital control subsystem (AOCS) of low-earth orbit satellite at Pakistan's space agency SUPARCO during 2011–2015. His research interests include linear and nonlinear control systems, embedded systems, and networked control systems.



**Dr. Almuatazzbellah M. Boker** is a Collegiate Assistant Professor at the Bradley Department of Electrical and Computer Engineering, Virginia Tech. He received his B.Eng in Mechatronics Engineering from the University of Leeds, UK, in 2002, his M.S. in Control Systems Engineering from the University of Sheffield, UK, in 2003, and his Ph.D. in Electrical Engineering from Michigan State University in 2013. Dr. Boker worked as an Assistant Lecturer at the Electrical and Electronics Engineering Department, University of Benghazi, Libya, 2004–2008; as a Post Doctoral Research Scholar at the NSF FREEDM Center, part of the Electrical and Computer Engineering Department, North Carolina State University, 2014–2015; and as an Assistant Professor at the Computer Sciences and Electrical Engineering Department, Marshall University, 2016–2019. Dr. Boker is an expert on the analysis and control of nonlinear systems. His research interests include data-driven estimation and control of dynamic systems. Dr. Boker is an Associate Editor of the IEEE Control Systems Society Conference Editorial Board. He received the 2002 Smallpeice Trust Prize from the University of Leeds for “an outstanding work on mechatronics” and the best poster award, 2012, Graduate Research Symposium, Michigan State University.



**Dr. Yousef Sardahi** is an Assistant Professor in the Weisberg Department of Mechanical Engineering at Marshall University. He earned his Ph.D. from the Department of Mechanical Engineering at the University of California, Merced, in 2016. His research interest includes Control System Design and Multi-Objective Optimization. His teaching experience includes Control Systems, Digital Controls, Automation and Control, System Modeling, Advanced Vibrations, Mechatronics, Circuits and Instrumentations, and Mechanical Engineering Computations.

**How to cite this article:** M. Aafaque, A. Boker, and Y. Sardahi, *Three-time scale multi-objective optimal PID control with filter design*, *Asian J Control* (2023), 1–13. <https://doi.org/10.1002/asjc.3060>