Hindawi
Modelling and Simulation in Engineering
Volume 2018, Article ID 2453817, 17 pages
https://doi.org/10.1155/2018/2453817

Research Article

Design of Optimal QFT Controller and Prefilter for Buck Converter Using Metaheuristic Algorithms

Nitish Katal and Shiv Narayan

Electrical Engineering Department, Punjab Engineering College (Deemed to Be University), Chandigarh, India

Correspondence should be addressed to Nitish Katal; nitishkatal@gmail.com

Received 28 August 2018; Accepted 8 November 2018; Published 2 December 2018

Academic Editor: Ricardo Perera

A buck converter is a step-down switching regulator. Buck convertors are being widely used in industrial applications that rely on regulated output voltage under fluctuating input voltage. A buck convertor works in the following modes: (a) current-controlled or (b) voltage-controlled mode. But these convertors manifest several nonlinearites because of the switching operation. Hence, in order to generate a quality output of the convertor, the design of a controller becomes crucial. In this paper, the synthesis of a QFT-based robust controller and prefilter has been carried out for an uncertain buck converter with varying input voltage and varying load. The controller synthesis problem has been posed as an optimization problem, and metaheuristic algorithms have been used for obtaining the optimal gains for the QFT controller and prefilter. By doing this, the QFT synthesis can be carried out in a single step instead of following the sequential classical QFT process on Nichols charts and the need for the generation of templates and bounds has be eliminated. The designed 2-degree-of-freedom QFT control system offers a robust behavior and efficiently handles the parametric uncertainties. The robustness of the designed controller has been confirmed through simulation results for large input voltage and load fluctuations.

1. Introduction

Buck converters are center to various diverse applications that require a tunable/fixed DC supply form a fixed/tunable DC supply such as aerospace, instrumentation, medical appliances, and computers. [1]. The regulation in such convertors is accomplished using pulse width modulation (PWM). The effect of such nonlinear switching and operational losses due to the continuous operation is often not considered in the controller synthesis process. Thus, modeling of plant ignorance, parametric uncertainty, nonlinearities, etc. in control system synthesis becomes crucial, so that the designed converter should assure quality output when operated for a long time, despite uncertainties in the system or any load variations etc. [2].

In the literature, several classical control methods have been implemented for the linearized buck convertor models but under the influence of uncertainties the performance of the convertors degrades [3]. Over years to mitigate the influence of such uncertainties, many robust control methodologies like H_2, $H\infty$, and μ-synthesis were established, and still these established theories ignored do not address the modeling of uncertainties in the plant model during the design process [4].

In 1960s, Issac Horowitz introduced Quantitative Feedback Theory (QFT). QFT has 2-degree-of-freedom (2-DoF) controller architecture, viz., (a) controller $K(s)$ and (b) prefilter $F(s)$, and the effect of closed loop uncertainties is reduced by the feedback controller while the feedforward prefilter shapes the desired frequency response. The QFT design process consists of several sequential steps, and the controller and prefilter is designed using loop shaping on Nichols charts. The loop-shaping process requires a lot of experience, and still there is no guarantee that an optimal controller has been designed.

This paper introduces an automated single-step QFT controller synthesis technique for buck convertors using metaheuristic algorithms. The desired QFT bounds and

performance objectives for the voltage mode control in the buck convertor have been expressed in terms of design objectives and constraints. The QFT controller system design problem has been expressed as an optimization problem. This eliminates the generation of the templates and bounds which else are required for the manual loop shaping. The designed control system offers a robust response over a range of parametric uncertainty both in frequency and time domain and also offers performance robustness for large input voltage variations. The work has also been compared with classical controller synthesis methods like Ziegler–Nichols, internal mode control (IMC), and the QFT controller proposed by Ibarra et al. [5] using the classical QFT design approach.

The paper has been split into subsequent sections: in Section 1, the ripples and other factors that limit the performance of buck convertors and how the QFT controllers can be beneficial are discussed. In Section 2, state-of-the-art literature has been reviewed. In Section 3, modeling of the buck convertor has been discussed. In Section 4, basics of QFT and the metaheuristic algorithms have been discussed. In Section 5, QFT design requirements for the buck convertor have been discussed, followed by the synthesis of the QFT controller for buck convertor using metaheuristic algorithms in Section 6. Results have been discussed in Section 7. In Section 8, the designed QFT controller has been used for the validation of the buck convertor for varying input voltages followed by conclusions and references.

2. Literature Review

Often, modeled plant dynamics have a lot of assumptions, and the operation of the plant and due to the aging of the instruments overtime lead to the deviation of the plant's nominal dynamics that were used while controller design. This makes it difficult to assure quality control over time. To address this issue, several control theories like $H\infty$, H_2, LQR, and μ-synthesis have been established, so that the designed control system must not deviate from its state in case of uncertainties [4]. But these control theories ignore the fact that the model used in controller synthesis is just the inexact model of the real plant [4].

In 1960s, Issac Horowitz introduced a frequency-domain controller design technique of Quantitative Feedback Theory (QFT) [4] based on Bode's gain-phase integrals. QFT's foundation is laid on the shaping of the feedback such that desired limits of robust stability, reference tracking, and disturbance rejection are satisfied. QFT has a 2-DoF control configuration, i.e., has a controller K(s) in the feedback loop and a prefilter F(s) prior the loop in feedforward configuration. The feedback controller K(s) mitigates the effect of closed loop variabilities, and the prefilter is designed to shape the input in such a way that a desired output is obtained both in time and frequency domains.

QFT has found application is several diverse engineering applications [6–14]. In these applications, manual loop shaping has been used for the synthesis of the QFT controller and prefilter. Manual loop shaping is carried out on Nichols charts and requires a lot of experience; the success of the design solely depends on the expertise of the engineer, and it is very hard to synthesize controllers for uncertain and nonminimum phase systems with complicated characteristics.

Recently, many researchers have emphasised on the automatic synthesis of QFT controllers. Gera and Horowitz [15] followed by Ballance [16] introduced a semi-iterative process for QFT controller synthesis, but very high-order controllers were obtained. But these approaches [17–19] are based on several unrealistic assumptions or a very conservative design as they solve the complicated nonlinear problem with complex or linear programming. Zolotas and Halikias [20] used the approach of obtaining the optimal QFT controllers by searching among the dense set of controllers. Patil et al. [21] automated the loop shaping procedure by translating the QFT design requirements and used the interval constraint satisfaction technique (ICST) for the automation process [22–27]. The controllers designed with the ICST-based approach suffer from overdosing over the frequency range.

Metaheuristic algorithms are now been widely used for the design of control systems. QFT control synthesis cannot be accomplished by using conventional gradient-based optimization algorithms. Several evolutionary algorithms have been used for the synthesis of the QFT controllers. Gracia-Sanz et al. [28] and Chen et al. [29] implemented GA for the automatic loop shaping of QFT controllers. Evolutionary algorithms have been used by Kim and Chung [30]. Molins and Garcia-Sanz [31] obtained robustly stable QFT controllers using both genetic and evolutionary algorithms. A robust PID type QFT controller has been designed by Satpati et al. [32] using PSO for automating the loop shaping process for time-delay systems. Meng and Xue [33] also used PSO for designing the fractional order of QFT controllers for the nonminimum-phase hydrosystem [34, 35]. But, some of these still require the generation of templates and bounds on Nichols chart for design purposes.

Still the applications of such algorithms in electrical engineering particularly in power electronics are very limited. Olalla et al. [36] elucidated on the QFT-based robust controller design for direct drive multipole wind turbines. Anmol R. Saxena and Veerachary [9] employed QFT for designing robust voltage mode control for higher order boost DC-DC switching power convertors. Khodabakhshian and Hemmati [10, 37] put to use QFT for robust control of a power system stabiliser (PSS) and DVR for distribution systems. The design of robust decentralised multimachine power system stabilisers (PSSs) ensures damping electromechanical oscillations and enhances the power system stability. Igrec et al. [38] presented QFT-based robust velocity control for series wound DC motors. Alavi and Saif [11] implemented a QFT-based robust control of integrated fault detection and control.

For the buck convertor, QFT has been employed for tackling the parametric uncertainty. In [5], classical QFT has been used for designing the robust controller. In [39], the bat algorithm has been used for the design of the robust $H\infty$ controller. In [40], the efficacy of the QFT-based PID controller over a Ziegler–Nichols-tuned PID controller has

been demonstrated. Still there is a lot of scope for minimizing the current and voltage ripples in the buck convertor, and QFT offers a promising solution to the problem.

3. Modeling of the DC-DC Buck Converter

The dynamics of the DC-DC convertor has been by small-signal state-space averaging to obtain a set of time-invariant equations [5]. Figure 1 shows the circuit diagram for a conventional buck regulator.

A buck convertor primarily operates in two configurations: (a) continuous current mode (CCM) and (b) discontinuous current mode (DCM). In this work, the continuous current mode (CCM) mode has been considered for the controller synthesis. For a duty cycle, equation (1) gives the current $G_{iD}(s)$ transfer function and equation (3) gives the voltage $G_{VD}(s)$ transfer function. Table 1 shows the parameters of the elements used in this paper:

$$G_{iD}(s) = \frac{V_{IN}\left(s\left(CR + Cr_C\right) + 1\right)}{CLR \cdot s^2 + A \cdot s + \left(R + r_L + r_s\right)}, \quad (1)$$

where

$$A = \left(L + CRr_L + C\left(Rr_s + r_C\left[R + L_S + r_L + r_s\right]\right)\right), \quad (2)$$

and

$$G_{VD}(s) = \frac{V_{IN} \times Rr_C\left(s\left(CR + Cr_C\right) + 1\right)}{A \cdot s^2 + B \cdot s + \left(R + r_C\right) \cdot \left(R + r_L + r_s\right)}, \quad (3)$$

where

$$A = \left(R + r_C\right) \cdot \left(CLR + CLr_C\right),$$
$$B = \left(R + r_C\right) \cdot \left(L + CRr_C + Cr_Cr_L + Cr_Cr_s + CR\left(r_L + r_s\right)\right). \quad (4)$$

4. Background

4.1. Quantitative Feedback Theory. Quantitative Feedback Theory (QFT) is a frequency-domain controller synthesis methodology introduced by Issac Horowitz in 1960s. QFT is based on Bode's famous gain-phase integrals and has a 2-DoF controller configuration as shown in Figure 2, a controller $K(s)$ which minimizes the effect of external disturbances, and a prefilter $F(s)$ which is used to shape the desired frequency response. In QFT, templates portray the degree of unpredictability in the plant on z-plane and the required performance indices in the form of bounds. The practicality of the controller design is measured using templates, and bounds provide the blueprint for shaping the open-loop transmission on the Nichols chart at each design frequency of interest. The controller is designed by shaping the open-loop transmission on the Nichols charts such that all the bounds are satisfied at each frequency of interest.

4.2. Bat Algorithm. In 2010, Yang proposed a metaheuristic algorithm of the bat algorithm (BA), which was established upon the echolocation behavior of the bats [41]. Bats use echolocation to search their prey and also to categorize distinct types of insects at night. An initial population of bats

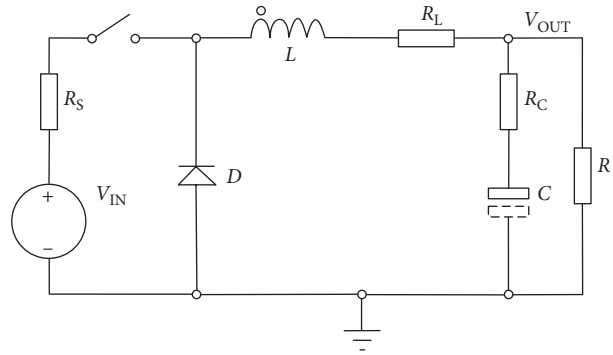

Figure 1: Conventional buck convertor.

Table 1: Values of the elements used in the DC-DC buck convertor.

Parameter	Symbol	Value
Input voltage	V_{IN}	24 V
Inductor	L	300 μH
Capacitor	C	220 μF
Load	R	12 Ω
PWM period	T_s	10 μs
PWM duty cycle	D	1
tON switch resistance	r_s	0.01 Ω
Inductor resistance	r_L	16.3 mΩ
Capacitor resistance	r_C	0.305 Ω

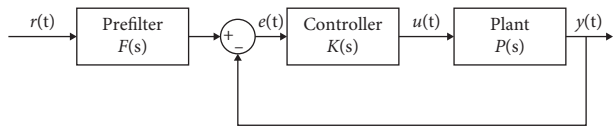

Figure 2: 2-Degree-of-freedom QFT control configuration.

is generated and has their respective positions x_i. Mathematically, the motion of the flying of the bat is given as

$$f_i = f_{min} + \left(f_{max} - f_{min}\right) \cdot \beta,$$
$$v_i^{t+1} = v_i^t + \left(x_i^t - x^*\right) \cdot f_i, \quad (5)$$
$$x_i^{t+1} = x_i^t + v_i^t,$$

where x_i is the position of the bat, v_i is the velocity by which the bat is moving and the updated positions x_i^t after the flight and velocities v_i^t at time t, uniformly distributed $\beta \in [0, 1]$ random vector, x^* is the current best global location for all n bats in the population. Initially, each bat is assigned with uniformly distributed frequency $[f_{min}, f_{max}]$.

When prey is identified by a bat, the loudness A_i is updated and the rate by which the pulse is being emitted r_i increases. Here, the initial loudness A_0 is 1, and a prey is identified by the bat, $A_{min} = 0$:

$$A_i^{t+1} = \alpha \cdot A_i^t,$$
$$r_i^t = r_i^0\left(1 - e^{-\lambda \cdot t}\right), \quad (6)$$

α and γ are fixed, and for $0 < \alpha < 1$, $\gamma > 0$:

$$A_i^t \longrightarrow 0,$$
$$r_i^t \longrightarrow r_i^0, \quad (7)$$
$$\text{as} \quad t \longrightarrow \infty.$$

The values of loudness and pulse emission rates are modified only when improved new solutions are obtained.

4.3. Flower Pollination Algorithm. Flower pollination algorithm is inspired by the occurrence of pollination that occurs in flowering plants [42] and is a medium of procreation in plants. It has been introduced in 2012 by Yang. In pollination, the pollens are transferred through pollinators like wind, birds, insects, animals, and other mediums. The fertilization of a flower in pollination can happen via self-pollination or allogamy. When the fertilization of the flower happens either from itself or from the different flower from the same plant, it is remarked as self-pollination, and if the pollens are from the different plant, then in that case, it is termed as allogamy or cross pollination.

Following four steps in pollination characterize the flower pollination algorithm [42]:

(1) For the pollination to occur from one flower to other, the pollinators perform Lévy flights and are regarded as global pollination

(2) Self-pollination is regarded as local pollination

(3) The more fertile the flower is, the greater the chances of the reproduction probability are

(4) The rate of global and local pollination is controlled by a switching probability, $p \in [0, 1]$

In global pollination, pollens are transmitted via pollinators, and the fertility of the flower secures pollination and the selection of the most fertile flower. Mathematically, it is represented as

$$x_i^{t+1} = x_i^t + L\left(x_i^t - g_*\right), \quad (8)$$

where x_i^t is the i^{th} pollen at iteration t and g_* is the present best solution at the current iteration. Pollination is controlled by L and is the step size. So, pollinators have to travel over an extent by performing Lévy flights. So, Lévy distribution is drawn as $L > 0$ and is given as follows:

$$L \sim \frac{\lambda\Gamma(\lambda)\sin(\pi\lambda/2)}{\pi}\frac{1}{s^{1+\lambda}}, \quad (s >> s_0 > 0), \quad (9)$$

where $\Gamma(\lambda)$ is the standard gamma function and the Lévy distribution is acceptable for big flights with $s > 0$.

Mathematically, the flower constancy in self-pollination and is given as follows:

$$x_i^{t+1} = x_i^t + \varepsilon\left(x_j^t - x_k^t\right), \quad (10)$$

where x_j^t and x_k^t are pollens from distinct parents but belonging to identical species. Local random search is used if the p x_j^t and x_k^t are selected from the identical population and ε is chosen as uniform distribution, i.e., $\varepsilon \in [0, 1]$.

Predominantly, flower pollination happens both at local and global extent. In case of nearby flowers, local pollination is dominant. So, switching probability p controls whether the pollination will be a global or local one and is 0.8.

4.4. Artificial Bee Colony Algorithm. Artificial bee colony (ABC) algorithm has been introduced by Karaboga and Basturk in 2005 and is based on the intelligent behavior of honeybees [43]. ABC is based upon the wide search of bees performing for searching the food and has three important components: employed bees, onlookers, and scouts. An employed bee is the one that inspects the food source inspected by it earlier, onlooker bees wait in the dancing area before choosing to go in search for food, and the bees that perform a random search is known as scout. The ABC algorithm has been divided into the following phases.

4.4.1. Initialization Phase. In the initialization phase, each bee in the population is assigned with a random location x_i, given as follows:

$$x_i = l_i + \text{rand}(0, 1) * (u_i - l_i), \quad (11)$$

where x_i is the food source of the i_{th} bee, u_i is the upper bound of x_i bee, l_i is the lower bound of x_i, and rand $(0, 1)$ is any random number between 0 and 1.

4.4.2. Employer Bee Phase. In this stage, the employer bee phase will perform search for the food v_i, in the neighborhood of the food source x_i as per its pervious memory. It is given mathematically as follows:

$$v_i = x_i + \varphi_i(x_i - x_k), \quad (12)$$

where v_i is the new food item of the i_{th} employed bee, x_i is the position of the i_{th} bee, and φ_i is randomly chosen and lies between $[-1, 1]$.

4.4.3. Onlooker Bee Phase. In this phase, the onlooker bees dance in the waiting area in the hive to share information regarding the employer bees. Based upon the probability of food and distance from the hive, the onlooker bees make their decision and are given mathematically as follows:

$$P_i = \frac{\text{fit}(x_i)}{\sum_{k=1}^{\text{SN}} \text{fit}(x_k)}, \quad (13)$$

where P_i is the probability of selection of the source by the bee, SN is the total number of bees in the colony, and fit (x_i) is the fitness of the ith bee.

Based upon the information shared, the onlooker bee will search the neighborhoods and calculate its fitness. Comparing the fitness of the current position with the previous one, the onlooker bees choose the new position.

4.4.4. Scout Bee Phase. After a certain iterations/searches, if an employed bee does not change its position, it becomes a scout. Scout bees are limited to one in a current cycle and performs search for the new food sources. When a new food location is found, it stores that in its memory till maximum number of cycles has been reached.

4.5. Biogeography-Based Optimization. In 2008, Simon introduced a biogeography-based optimization algorithm [44], based upon the migration of species from one island to another. It is an evolutionary algorithm that leads to the generation of some new species and the extinction of some other. The relocation of species is governed by the habitat suitability index (HSI). A habitat/island with higher HSI is regarded as more suitable for living, and the lower HSI means that it is not suitable. Various features like vegetation, water, area, and temperature feature the habitat and are called as the suitability index variables (SIV). A habitat with high HSI has a higher emigration rate and lower immigration rate. Mathematically, it can be given as

$$\lambda = I \cdot \left(1 - \frac{S}{S_{\mathrm{MAX}}}\right), \qquad \mu = \frac{E \cdot S}{S_{\mathrm{MAX}}}, \tag{14}$$

where S is the number of species in equilibrium, S_{MAX} is the maximum number of species, μ is the emigration rate, and λ is the immigration rate.

The probability P_S changes from time t to time $(t + \Delta t)$ and is given mathematically as

$$P_S(t + \Delta t) = P_S\left(1 - \lambda_S \Delta t - \mu_S \Delta t\right) + P_{S-1}\lambda_{S-1}\Delta t + P_{S+1}\mu_{S+1}\Delta t, \tag{15}$$

where λ_S and μ_S are the immigration and emigration rates for S species in the habitat.

Two important operators: (a) migration and (b) mutation, govern the BBO algorithm. A habitat/island with higher HSI is regarded as more suitable for living, and the lower HSI means that it is not suitable. Migration is an adaptive activity. Probability P_{mod} is used to modify the information gathered from the species during the process of emigration and immigration to modify the SIV. Elitism is preserved in the BBO, to preserve the best solutions from being abandoned. Mutation refers to calamities that change the HSI and disturb the equilibrium of the species. So due to these calamities, the HSI of the habitat can abruptly change and is regarded as SIV mutation and the rate of mutation is dependent on the species count probability. Elitism is preserved in this stage too, in order to favour the best solutions. Mathematically, mutation rate is given as

$$m = m_{\max}\left(1 - \left(\frac{P_S}{P_{\mathrm{MAX}}}\right)\right), \tag{16}$$

where P_S is the probability of S species on a island, P_{MAX} is the maximum number of species, and $m_{\max}$ is the maximum mutation rate.

4.6. Harmony Search Algorithm. In 2000, Geem and Loganathan introduced a population-based algorithm which is based on the principles of the extemporization process in jazz instruments [45]. While composing a harmony, musicians try out various possible musical pitches they remember, so that by using an optimal combination of such pitches they compose a perfect harmony. Harmony search algorithm comprises the following steps:

(1) Initialization of the randomly generated harmony search memory (**HM**). For an n-dimensional problem, the solution space can be given as

$$\mathbf{HM} = \begin{bmatrix} x_1^1 & x_2^1 & x_3^1 & \dots & x_n^1 \\ x_1^2 & x_2^2 & x_3^2 & \dots & x_n^2 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ x_1^{\mathrm{HMS}} & x_2^{\mathrm{HMS}} & x_3^{\mathrm{HMS}} & \dots & x_n^{\mathrm{HMS}} \end{bmatrix}, \tag{17}$$

where $[x_1^1, x_2^1, x_3^1, \dots, x_n^1]$ $(i = 1, 2, 3, \dots, \mathbf{HMS})$ is the solution vector.

(2) In this step, a now improved result $[x_1', x_2', x_3', \dots, x_n']$ is produced from the HM. Harmony memory considering rate (HMCR) controls the probability of the selection of a HM, and the pitching adjust rate (PAR) controls the probability of a HM to be mutated. These parameters of HMCR and PAR mimic the crossover and mutation like that of GA, but in GA, there is a limitation on the parents in the selection phase, while in the harmony search, the selection is across the array.

(3) In this step, the HM is updated. The fitness of the new solutions is evaluated, it returns a better value than the worst in the HM, and the worst one is replaced by the new solution. If not, the new solution is discarded.

(4) Repeat steps 2 and 3 till the stopping criterion is reached.

The harmony search algorithm has many operators like those in evolutionary algorithms, but harmony search differs from all as it offers single search memory for the solution to evolve. This also boosts the convergence speed of the algorithm.

4.7. Differential Evolution. Differential evolution (DE) [46] is a population-based stochastic direct search optimization algorithm and uses the operators of crossover, mutation, and selection. Mutation operator is the zeal of DE for producing better results, while in GA, crossover is used. In DE, the mutation operator is used for search and selection of the global best solution, and sometimes-scattered crossover is also used for generating better solutions [15]. In this paper, DE with jitter [17] has been utilized in the design process.

4.7.1. Mutation. A mutation vector is generated for each target vector $x_{i,G}$ a and is given as

$$v_{i,G+1} = x_{i,G} + K \cdot \left(x_{r1,G} - x_{i,G}\right) + F \cdot \left(x_{r2,G} - x_{r3,G}\right), \tag{18}$$

where $r_1, r_2, r_3 \notin \{1, 2, \dots, \mathrm{NP}\}$ are generated randomly and cannot be same, F is the scaling factor, and K is the combination factor.

4.7.2. Crossover. The trial vector $u_{ji,G+1}$ is generated by the crossover operator by blending the parent with the mutated vector and is given as

$$u_{ji,G+1} = \begin{cases} v_{ji,G+1} & \text{if } (\text{rnd}_j \le \text{CR}) \text{ or } j = rn_i, \\ q_{ji,G} & \text{if } (\text{rnd}_j > \text{CR}) \text{ or } j \ne rn_i, \end{cases} \quad (19)$$

where $j = 1, 2, \ldots, D$, random vector $r_j \in [1, 0]$, crossover constant $\text{CR} \in [1, 0]$, and randomly chosen index $rn_i \in (1, 2, \ldots, D)$.

4.7.3. Selection. In selection, any individual from the population can form the parent despite of its fitness. After mutation and crossover, the competency of the child is assessed and equated with the competency of the parent and the individual with a better competency value is chosen.

4.8. Imperialist Colony Algorithm. Imperialist colony algorithm is inspired by the imperialist competition [46]. Initially, a population is generated, and each member is called as country and is divided into two types: (a) colonies and (b) imperialists. These empires compete with each other; the weakest ones fall, and the powerful ones take the possession of their colonies; and this lays the foundation of this algorithm. At the end of the competition, only one imperialist survives, with all the colonies having the same cost of the imperialist. The colonies then start to move towards their imperialist empires and follow a simple model of assimilation policy. α and x are uniformly distributed random numbers and are given as

$$\begin{aligned} x &\sim U(0, \alpha \times d), \\ \alpha &\sim U(-\gamma, \gamma), \end{aligned} \quad (20)$$

where α and γ are random numbers and are used to modify the area of the empire.

Both the power of the imperialist and its colonies together form the strength of the nation. The empire that fails to adhere with the competition becomes extinct. The competition mainly strengthens the empire and decreases the power of the weak nations and makes them extinct. The competition amongst empires is a way to converge them towards a single powerful empire in the world and with all the other countries as its colonies.

4.9. Invasive Weed Optimization. In 2006, Mehrabian introduced the invasive weed optimization (IWO) algorithm [47], which mimics the spreading strategy of the weeds. Weeds are regarded as unwanted plants and are very tough and adjustable which makes them very unwanted in farming. The IWO algorithm uses the common operators of seeding, growth, and competition. The algorithm consists of following main phases:

4.9.1. Initialization. In this phase, a population of weeds is randomly generated.

4.9.2. Reproduction. In this phase, only a few plants in the population produce seeds and this depends on the fitness of the plant. The plant, which is least fit, will produce lesser seeds, while the fittest one will produce the most number of seeds, and this relation is linear.

4.9.3. Spatial Distribution. In this phase, a random dispersion of the seeds is carried out such that the seeds remain nearer the parent. As the generations pass by, the standard deviation σ of the random number is reduced from the initial σ_{initial} to the final σ_{final} in each step and is given mathematically as

$$\sigma_{\text{iter}} = \frac{(\text{iter}_{\max} - \text{iter})^n}{\text{iter}_{\max}^n} (\sigma_{\text{initial}} - \sigma_{\text{final}}) + \sigma_{\text{final}}. \quad (21)$$

4.9.4. Competitive Exclusion. If a weed plant fails to produce seeds, it will become extinct; otherwise, they would take over the world. So, competition limits the number of plants in the colony. As the generations pass, it is desired that the fitter plant reproduce more than the unfit ones. When a maximum number of weeds $P_{\max}$ in a colony is reached, the elimination of the unfit weeds kick in. The weeds with minimum fitness are eliminated, and new seeds are generated and dispersed by healthier plants using reproduction and spatial distribution, and at the end, the fitness is evaluated. The weeds with minimum fitness are eliminated, and the process goes on and on till the stopping criteria is met.

4.10. TLBO. In 2011, motivated by the process of teaching and learning, Rao et al. introduced teaching-learning-based optimization (TLBO) [48]. The essence of the TLBO is that the teacher influences the performance of the students in the classroom. The algorithm works in two stages: (a) teacher phase and (b) learner phase. Teacher is regarded the influencer, and students in the class can learn from the teacher and also from the interaction among themselves. The grades monitor the measure of learning and are directly influenced by the teacher. TLBO is a population-based algorithm, where n is the number of learners (population size), m is the different subject the pupil has to learn (the dimension of the problem), and the teacher is considered as the best solution amongst all solutions. Both the phases of TLBO are discussed as follows.

4.10.1. Teaching Phase. In the 1st phase of TLBO, the students learn form the teacher. The teacher is regarded as the elite being and shares his expertise with the students to increase their knowledge (the mean result). Initially, a random population is generated, and the individual with the minimum fitness value is chosen as a teacher (for minimization problems), and this information is shared with the students to increase their mean scores from M_A to M_B. The teacher tries to increase the mean of the class, by bringing it closer to its value, but it is also dependent on the capability of learners.

Suppose M_I be the mean result and teacher be T_i at any iteration i. The teacher T_i will try to improve the mean by converging it towards its own level, and the new mean is given by M_{new} [7] and is given by

$$\text{Diff\_Mean}_i = r_i\left(M_{\text{new}} - T_{\text{F}}M_i\right), \tag{22}$$

where T_{F} is the teaching factor and r_i is any random number between [0, 1].

Teaching factor is limited to 1 or 2 only and is chosen by T_{F} with equal probability:

$$T_{\text{F}} = \text{round}\left[1 + \text{rand}\,(0,1)\{2-1\}\right]. \tag{23}$$

The new solution X_{new} is generated by adding the difference mean as given by

$$X_{\text{new},i} = X_{\text{old},i} + \text{Diff\_Mean}_i. \tag{24}$$

4.10.2. Learner Phase. In the learning phase, the pupils learn from mutual interaction. The interactions are random and happen if and only if the grade of one student is larger than other. For two learners X_i and $X_j\,(i \neq j)$, the mathematical expression for the learning phase is as

$$\begin{aligned} X_{\text{new},i} &= X_{\text{old},i} + r_i\left(X_i - X_j\right) \text{ if } f\left(X_i\right) < f\left(X_j\right), \\ X_{\text{new},i} &= X_{\text{old},i} + r_i\left(X_j - X_i\right) \text{ if } f\left(X_i\right) > f\left(X_j\right). \end{aligned} \tag{25}$$

The new solution is accepted only when it minimizes or maximizes the objective function. As the teaching-learning process progresses, the level of knowledge of learners increases towards to that of the teacher and the algorithm converges towards a solution.

4.11. Ant Colony Optimization. Ant colony optimization is a probabilistic metaheuristic algorithm inspired by the behavior of ants for finding the optimal path from their colony to the food source [49]. Each ant lays a trail of pheromones, which act as a guide for the following ant. The algorithm has three main steps: initialization, formulation of the ant solution, and upgrading the pheromone trail.

The global ant system updates the pheromone trail, and all the ants in the colony have to share the information of their journeys and the deposition of the pheromone. Mathematically, it is given as

$$\tau_{ij}(t+1) = (1-\rho)\tau_{ij}(t) + \sum_{\text{K}} \frac{Q}{L_{\text{K}}}, \tag{26}$$

where τ_{ij} is the probability between town i and j, Q is taken as a constant, L_{K} is the length of the tour by the K_{th} ant, and ρ is the evaporation rate of the pheromone.

5. QFT Design Requirements

QFT controller is primarily implemented to mitigate the consequences of parametric variations of the uncertain dynamics of the plant. The synthesis of the controller $K(s)$ and prefilter $F(s)$ is carried out by shaping the open-loop transmission transfer function $L_0(s) = K(s)G_0(s)$ on the Nichols charts such that a set of predefined performance objectives are met. These predefined performance objectives form the bounds that check the loop-shaping process. For designing the QFT controller for the buck convertor, the objectives of robust stability, tracking, and sensitivity have been considered in the design process [50]. The range of design frequencies considered in this paper is ω = [0.4, 0.8, 1.2, 1.7, 2.1, 10, 25, 50, 100, 200] rad/sec, and the designed QFT controller must satisfy the design requirements at each design frequency.

5.1. Robust Stability. For the closed loop system to assure robust stability, the minimization of the maximum magnitude of the closed-loop frequency response of the closed-loop system at each design frequency is desired. Mathematically, it is given as

$$\left|\frac{L(j\omega)}{1+L(j\omega)}\right| \leq \delta_1, \tag{27}$$

where $L(j\omega) = K(j\omega)G_0(j\omega)$ is the open-loop transfer function and δ_1 is a constant.

Equation (28) gives the maximum variability of magnitudes at each ω_i for the uncertain plant $G(j\omega)$ and is given by $\delta_P(j\omega_i)$. Equation (29) gives the difference between the upper and lower tracking bounds:

$$\delta_P(j\omega_i) = \left|G(j\omega_i)\right| - \left|G_0(j\omega_i)\right|, \tag{28}$$

$$\delta_{\text{R}}(j\omega_i) = \left|T_{\text{U}}(j\omega_i)\right| - \left|T_{\text{L}}(j\omega_i)\right|. \tag{29}$$

$\delta_{\text{L}}(j\omega_i)$ gives the maximum variation of magnitude of the closed-loop system in equation (30). For the system to be robustly stable, the minimization of $\delta_{\text{L}}(j\omega_i)$ has been considered, it must follow the following constraint, and $\delta_{\text{L}}(j\omega_i) < \delta_{\text{R}}(j\omega_i)$:

$$\delta_{\text{L}}(j\omega_i) = \left|\frac{K(j\omega_i)\cdot G(j\omega_i)}{1+K(j\omega_i)\cdot G(j\omega_i)}\right| - \left|\frac{K(j\omega_i)\cdot G_0(j\omega_i)}{1+K(j\omega_i)\cdot G_0(j\omega_i)}\right|. \tag{30}$$

5.2. Tracking Performance. Tracking ratios guide the shaping of the open-loop transmission, such that a set of time- and frequency-domain specifications is satisfied. The upper and lower tracking ratios are declared at the starting of the design process. Mathematically, it is given as in the following equation:

$$\left|T_{\text{L}}(j\omega)\right| \leq \left|\frac{F(j\omega)L(j\omega)}{1+L(j\omega)}\right| \leq \left|T_{\text{U}}(j\omega)\right|. \tag{31}$$

Upper $T_{\text{U}}(j\omega)$ and lower $T_{\text{L}}(j\omega)$ bounds are given by equations (8) and (9), respectively, as

$$T_{\text{U}}(j\omega) = \frac{2.95\times 10^9}{s^2 + 5.4\times 10^5 \cdot s + 2.95\times 10^9}, \tag{32}$$

$$T_{\text{L}}(j\omega) = \frac{1.48\times 10^{12}}{s^3 + 63.6\times 10^3 \cdot s^2 + 5.89\times 10^8 \cdot s + 1.48\times 10^{12}}. \tag{33}$$

Minimization of $\delta_F(j\omega_i)$ has to be carried out at each design frequency, so that the tracking bounds are satisfied and is given mathematically as

$$\delta_F(j\omega_i) = \left|\frac{T_U(j\omega_i) - T_L(j\omega_i)}{2}\right| - \left|\frac{F(j\omega_i)\cdot L(j\omega_i)}{1 + L(j\omega_i)}\right|. \quad (34)$$

5.3. Sensitivity. The designed system must be immune to external disturbances. So, minimization of the sensitivity ensures that and is given mathematically as

$$J_S = |S(j\omega_i)|, \quad (35)$$

where

$$S(j\omega_i) = \frac{1}{1 + G(j\omega_i)K(j\omega_i)}. \quad (36)$$

6. Synthesis of the QFT Controller and Prefilter Using Metaheuristic Algorithms

The QFT controller and prefilter must satisfy the design specifications of robust stability, tracking performance, and sensitivity. The QFT controller synthesis problem has been expressed as an optimization problem which offers a templates-and-bounds-free approach for designing optimal QFT controllers within very less time and also naïve loop-shaping experience. In this paper, a standard PID controller and a fixed structure prefilter have been chosen and are given by equations (37) and (38) respectively:

$$K(s) = K_P + \frac{K_I}{s} + K_D \cdot s \quad (37)$$

$$F(s) = \frac{a}{b \cdot s + a} \quad (38)$$

Algorithms mentioned in Section 4 have been used to solve the QFT design problem. These algorithms aim at finding the optimum values for the controller, and the prefilter gains $[K_P, K_I, K_D, a, b]$ such that the predefined QFT objectives are satisfied. Equation (39) gives the QFT controller synthesis objective function, which has been expressed as aggregate of function:

$$J = \alpha_1 \cdot \delta_L(j\omega_i) + \alpha_2 \cdot \delta_F(j\omega_i) + \alpha_3 \cdot J_S, \quad (39)$$

where the values of α_1, α_2, and α_3 have been carefully set to [1, 100, 100] based upon several trials.

7. Results and Discussions

In this paper, the voltage mode-controlled DC-DC buck convertor is considered. The parameters of the physical components in Table 1 have been used to derive the nominal plant transfer function, given by equation (40). The proposed objective function J given by equation (39) has been minimized using the metaheuristic algorithm:

$$G_0(s) = \frac{s + 3.54 \times 10^8}{s^2 + 1415.19 \cdot s + 1.48 \times 10^7}. \quad (40)$$

The optimal QFT controller and prefilter obtained from the automated synthesis are given by K_{BA} and F_{BA} are the controller and prefilter obtained by the bat algorithm, K_{FPA} and F_{FPA} are the controller and prefilter obtained by the flower pollination algorithm, K_{BBO} and F_{BBO} are the controller and prefilter obtained by the biogeography-based optimization algorithm, K_{HS} and F_{HS} are the controller and prefilter obtained by the harmony search algorithm, K_{DE} and F_{DE} are the controller and prefilter obtained by differential evolution, K_{ICA} and F_{ICA} are the controller and prefilter obtained by the imperialist colony algorithm, K_{IWO} and F_{IWO} are the controller and prefilter obtained by the invasive weed optimization algorithm, K_{TLBO} and F_{TLBO} are the controller and prefilter obtained by the teaching-learning-based optimization algorithm, and K_{ACO} and F_{ACO} are the controller and prefilter obtained by the ant colony optimization algorithm.

To compare the designed QFT-based controller and prefilter, the results have been compared with several classical controller synthesis methodologies. K_{ZN} gives the transfer function of the controller obtained using the classical PID controller design method of Ziegler–Nichols. K_{IMC} gives the transfer function of the controller obtained using classical IMC-based controller synthesis, and $K_{LIbarra}$ and $F_{LIbarra}$ gives the transfer function of the QFT controller obtained [5] using the classical QFT synthesis process and is a 3rd order controller with very large gain values:

$$K_{BA}(s) = 207.69 + \frac{854.89}{s} + 15.202 \cdot s,$$

$$F_{BA}(s) = \frac{3220.644}{0.877 \cdot s + 3220.644},$$

$$K_{FPA}(s) = 106.613 + \frac{1320}{s} + 9.791 \cdot s,$$

$$F_{FPA}(s) = \frac{3976.568}{1.084 \cdot s + 3976.568},$$

$$K_{BBO}(s) = 1012.32 + \frac{8147.32}{s} + 100.32 \cdot s,$$

$$F_{BBO}(s) = \frac{3102.3}{1.27 \cdot s + 3102.3},$$

$$K_{HS}(s) = 118.899 + \frac{1364.655}{s} + 14.22 \cdot s,$$

$$F_{HS}(s) = \frac{4132.538}{1.09 \cdot s + 4132.538},$$

$$K_{DE}(s) = 98.4 + \frac{3007.43}{s} + 9.88 \cdot s,$$

$$F_{DE}(s) = \frac{3500.93}{1.2 \cdot s + 3500.93},$$

$$K_{ICA}(s) = 470.578 + \frac{1968.2}{s} + 26.396 \cdot s,$$

$$F_{ICA}(s) = \frac{3758.2}{1.47 \cdot s + 3758.2},$$

$$K_{\mathrm{IWO}}(s) = 286.88 + \frac{1480.6}{s} + 10 \cdot s,$$

$$F_{\mathrm{IWO}}(s) = \frac{4626.8}{1.1 \cdot s + 4626.8},$$

$$K_{\mathrm{TLBO}}(s) = 120.582 + \frac{1521.81}{s} + 48.364 \cdot s,$$

$$F_{\mathrm{TLBO}}(s) = \frac{4031.85}{1.11 \cdot s + 4031.85},$$

$$K_{\mathrm{ACO}}(s) = 9.7 \times 10^{8} + \frac{-6.39 \times 10^{10}}{s} + 1.303 \times 10^{12} \cdot s,$$

$$F_{\mathrm{ACO}}(s) = \frac{1179.27}{0.89 \cdot s + 1179.27},$$

$$K_{\mathrm{ZN}}(s) = 0.9374 + \frac{376.785}{s} + 5.83 \times 10^{-5} \cdot s,$$

$$K_{\mathrm{IMC}}(s) = 160.49 \frac{\left((0.00026 \cdot s)^2 + 9.6 \times 10^{-5} \cdot s + 1\right)}{2.8 \times 10^{-9} \cdot s^2 + s},$$

$$K_{\mathrm{LIbarra}}(s) = \frac{5.44 \times 10^{12} \cdot s^2 + 5.44 \times 10^{16} \cdot s + 1.142 \times 10^{20}}{2.1 \times 10^{7} \cdot s^3 + 3.675 \times 10^{12} \cdot s^2 + 7.875 \times \cdot 10^{16} s},$$

$$F_{\mathrm{LIbarra}}(s) = \frac{3450}{s + 3450}. \tag{41}$$

7.1. Nominal Case. Figures 3 and 4 show the compared closed-loop step and frequency response of the nominal system with the optimal controller and prefilter parameters obtained after optimization. The various time-domain performance of the system is given in Table 2. From Figures 3 and 4, it can be seen that the controller and prefilter designed using differential evolution, imperialist colony algorithm, biogeography-based optimization, and ant colony optimization fail to satisfy the design requirements in both time and frequency domains and hence are not suitable for application. Of all the designed QFT controllers, one derived from artificial bee colony optimization gives the best time response and also satisfies the frequency-domain specifications. Also, the Ziegler–Nichols-tuned controller offers a highly oscillatory response with an overshoot percentage of 51.3%. The QFT controller designed by Ibarra et al. [5] and IMC controller also offer satisfactory performance both in time and frequency domains.

7.2. Parametrically Uncertain Plant (Worst Case Response). So, to verify the robustness of the controller to parametric uncertainties, an uncertain buck convertor is considered as

$$G(\mathrm{s}) = \frac{\left[1.62 \times 10^{4}, 3.03 \times 10^{4}\right] \cdot s + \left[2.41 \times 10^{8}, 4.52 \times 10^{8}\right]}{s^2 + \left[1.14 \times 10^{3}, 3.88 \times 10^{3}\right] \cdot s + \left[1.22 \times 10^{7}, 1.5 \times 10^{7}\right]}. \tag{42}$$

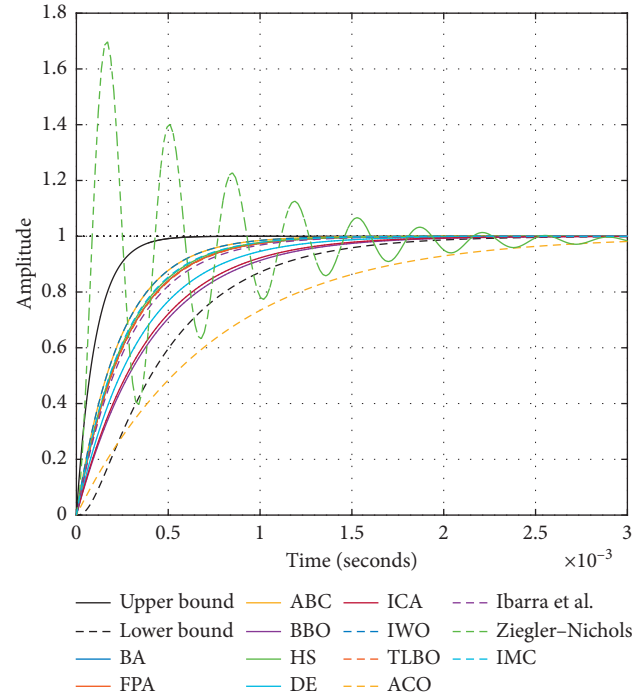

Figure 3: Compared step response of the closed-loop system (the nominal case).

The compared closed-loop step and frequency response of the parametrically uncertain system are shown in Figures 5 and 6. From the figures, it can be clearly concluded that the closed loop performance of the system lies within bounds and lies within the vicinity of nominal response. But the QFT controllers designed using differential evolution, imperialist colony algorithm, biogeography-based optimization, and ant colony optimization fail to satisfy the design requirements in both time and frequency domains. The controller designed using Ziegler–Nichols and the resultant controller offered very poor response with very high overshoot percentage of 51.3% which is not acceptable and can be seen in Figure 3. When this ZN-tuned controller is subjected to an uncertain plant, it showed a highly oscillatory response, as can be seen in Figure 5. Also, in frequency domain, the ZN-tuned controller offers a very poor response as can be seen in Figures 4 and 6. From Figures 5 and 6, it can be seen that the designed IMC-based controller offers a stable and less oscillatory response but fails to satisfy the tracking performances in time domain and frequency domain.

8. Design Validation

8.1. Variable Input Voltage. A buck convertor with varying input voltage has been designed in SIMULINK to test the efficacy of the designed QFT controllers and prefilters. For the ideal response, the buck convertor must maintain a fixed output voltage despite of the fluctuation in the input voltage. In this case, a fixed load of 50 Ω has been considered, while the input voltage has been varied from 20–28 V for a fixed

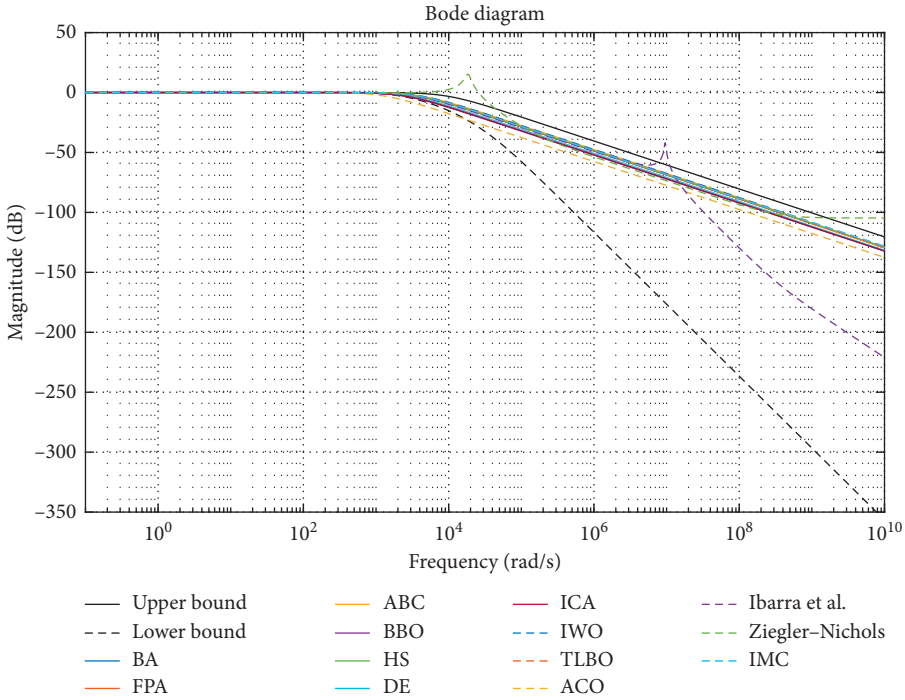

Figure 4: Compared frequency response of the closed-loop system (the nominal case).

Table 2: Time-domain performance indices.

Performance indices	Rise time	Settling time	Overshoot percentage (%)
BA	5.983×10^{-04}	0.0011	0
FPA	5.989×10^{-04}	0.0011	0
ABC	5.231×10^{-04}	9.315×10^{-04}	0
BBO	8.994×10^{-04}	0.0016	0
HS	5.795×10^{-04}	0.0010	0
DE	7.531×10^{-04}	0.0013	0
ICA	8.593×10^{-04}	0.0015	0
IWO	5.223×10^{-04}	0.9045	0
TLBO	6.048×10^{-04}	0.0011	0
ACO	0.0017	0.0030	0
ZN	1.83×10^{-04}	1.021	51.3
Ibarra	6.368×10^{-04}	0.0011	0
IMC	5.72×10^{-04}	0.001	0

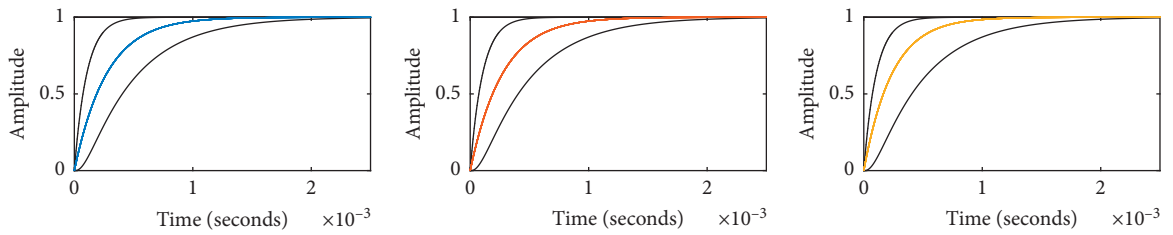

Figure 5: Continued.

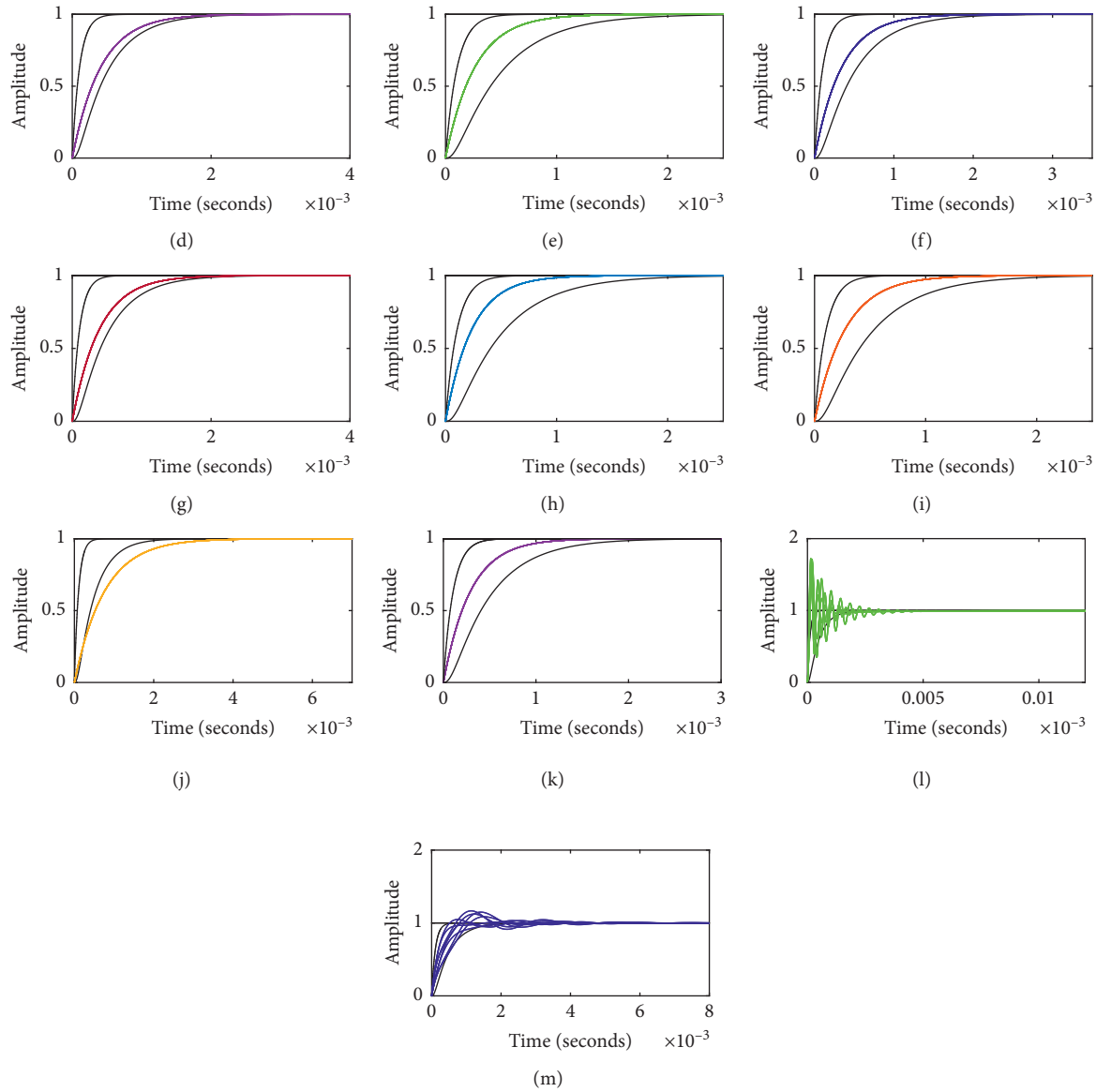

FIGURE 5: Compared time-domain worst case response of the closed-loop system: (a) BA; (b) FPA; (c) ABC; (d) BBO; (e) HS; (f) DE; (g) ICA; (h) IWO; (i) TBLO; (j) ACO; (k) Ibarra et al; (l) Zeigler–Nichols; (m) IMC.

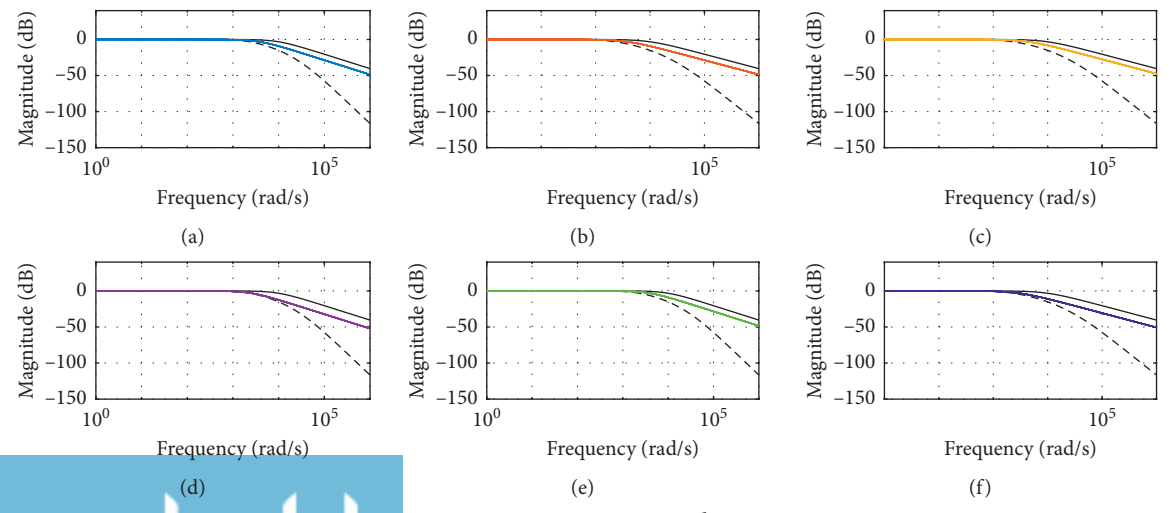

FIGURE 6: Continued.

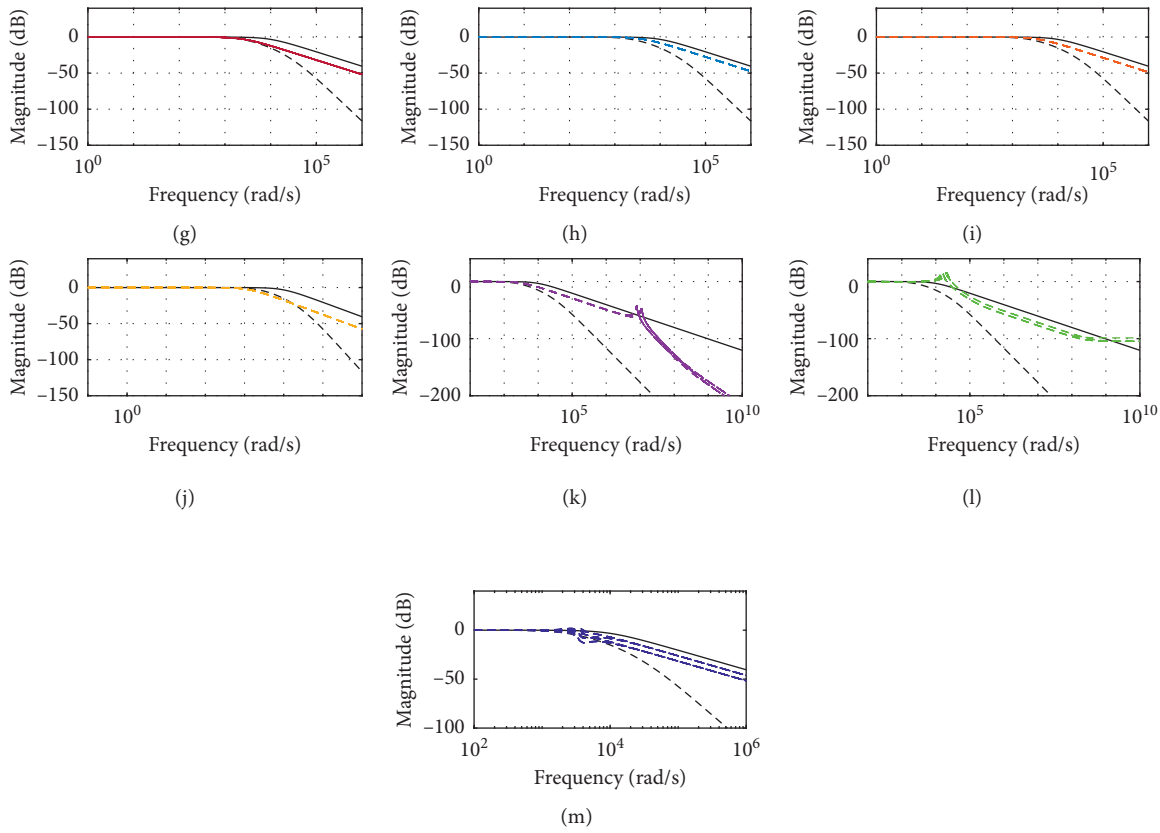

FIGURE 6: Compared frequency-domain worst case response of the closed-loop system: (a) BA; (b) FPA; (c) ABC; (d) BBO; (e) HS; (f) DE; (g) ICA; (h) IWO; (i) TBLO; (j) ACO; (k) Ibarra et al.; (l) Zeigler Nichols; (m) IMC.

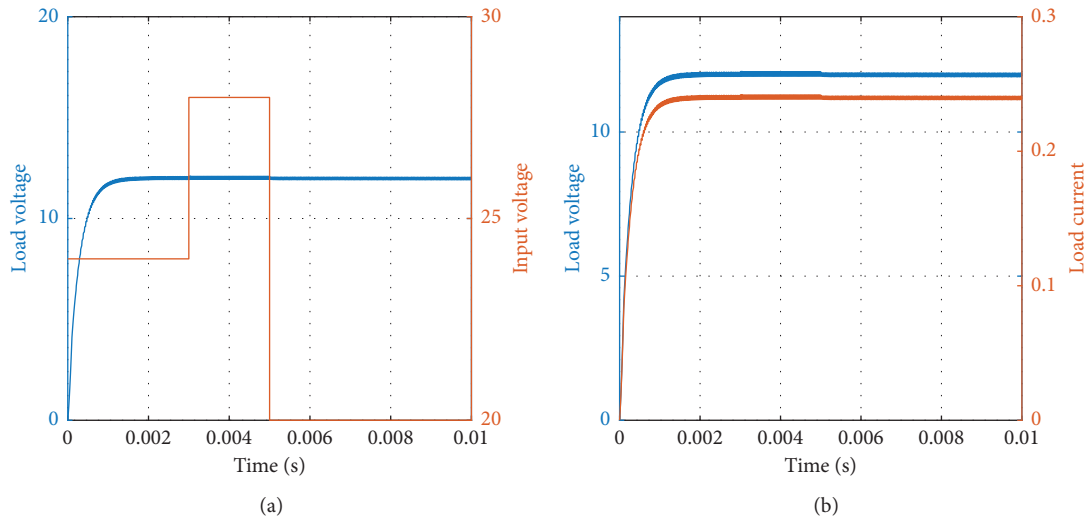

FIGURE 7: Plot for (a) output load voltage and the variable input voltage and (b) output current and voltage for varying input voltage case using flower pollination algorithm-tuned QFT controller.

output of 12 V. The plot for load voltage for variable input voltages has been shown in Figure 7(a). The plot for load voltage and load current is shown in Figure 7(b). From Table 3, it can be seen that the QFT controller designed using flower pollination algorithm offers the minimal current and voltage ripples, as can be seen in Figure 8, when compared to other metaheuristic algorithms and the time- and frequency-domain performances too lie in the design bounds.

Table 3: Convertor performance parameters (variable input voltage).

Parameters	Voltage ripple (ΔV) (%)	Current ripple (ΔI) (%)
BA	1.167	1.042
FPA	0.833	0.833
ABC	0.833	1.041
BBO	1.083	1.375
HS	1.167	1.25
DE	0.833	1.041
ICA	1.041	1.041
IWO	1.083	1.375
TLBO	1.041	1.041
ACO	1.041	1.041
ZN	0.477	0.333
IMC	0.375	0.479
Ibarra	0.67	0.67

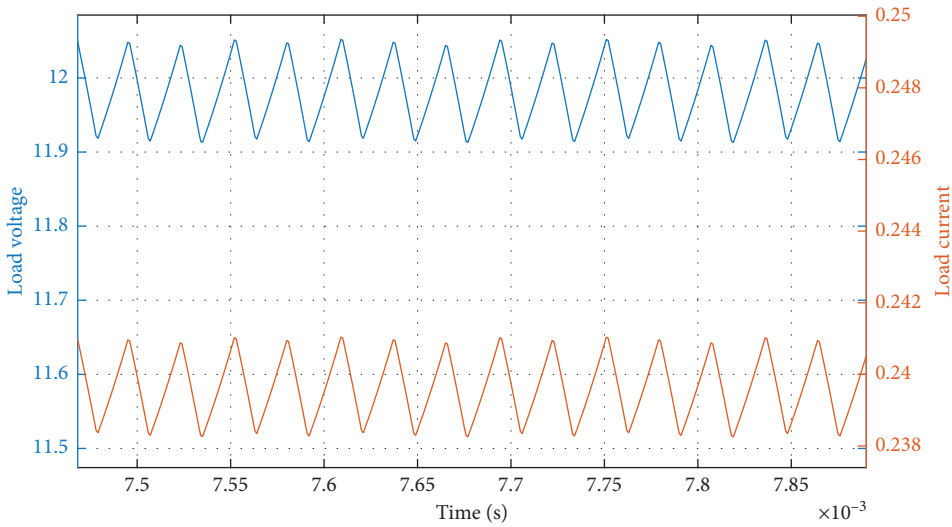

Figure 8: Plot for output current and voltage ripple for varying input voltage case using flower pollination algorithm-tuned QFT controller.

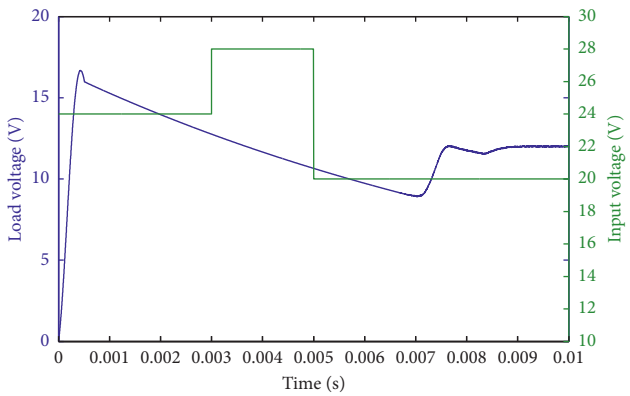

Figure 9: Output voltage across load under variable input voltage for ZN-PID controller.

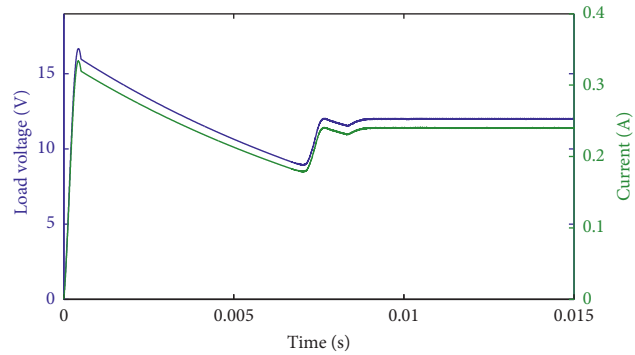

Figure 10: Output current and voltage for varying input voltage for ZN-PID controller.

The plot for load voltage for variable input voltages and the plot for load voltage and load current are shown in Figures 9 and 10, and it can be seen that the Ziegler–Nichols-tuned PID controller offers a very poor response and fails to offer a sustained DC output under variations in input voltage. Also, in Figure 11, it can be seen that the designed IMC controller too offers a poor response as it fails to offer a stable output in case of varying load changes. Figure 12 shows the plot for load voltage for variable input voltages and the plot for load voltage and load current for the QFT controller and prefilter designed by Ibarra et al. [5] and only attains an average voltage output of 11.8 V.

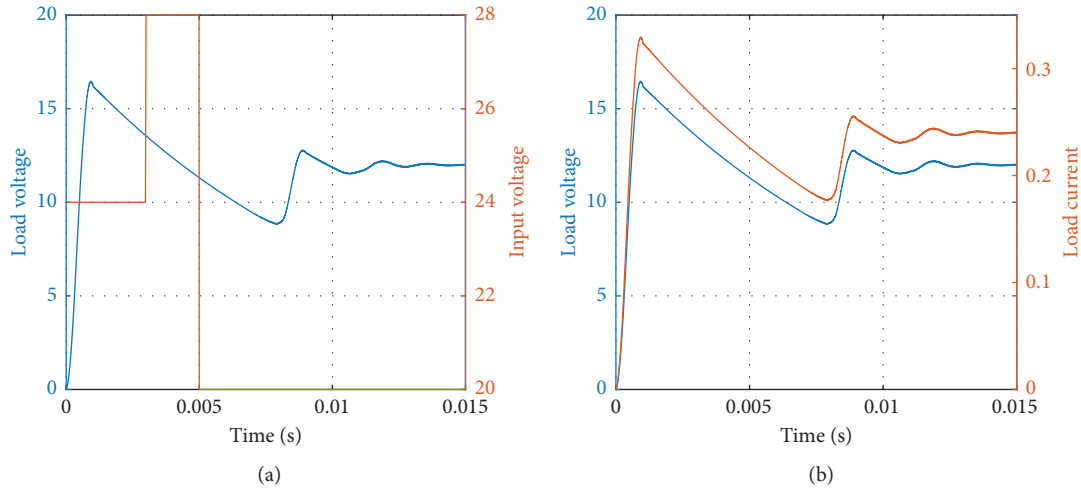

Figure 11: Plot for (a) output load voltage and the variable input voltage and (b) output current and voltage for varying input voltage case using the IMC controller.

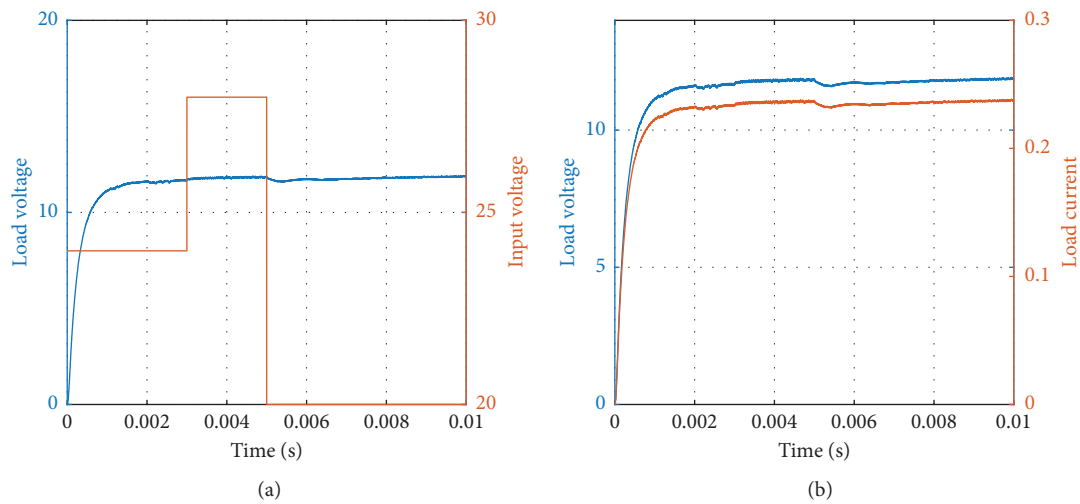

Figure 12: Plot for (a) output voltage across load under variable input voltage and (b) output current and voltage for varying input voltage using the QFT controlled proposed by Ibarra.

8.2. Variable Load. To check the efficacy of the designed robust QFT-based control scheme, the buck convertor with varying load has been considered. Here, the input voltage has been fixed at 48 V, for obtaining a fixed output voltage of 24 V. The variation in the resistive output load has been considered and has been varied form 2 Ω to 57 Ω. Figure 13 shows the simulation results for the output voltage as the load is varied from 2 Ω to 57 Ω, and it can be seen that, as the load changes, the variation in load current can be witnessed while the designed the designed QFT-based control system successfully maintains a constant output voltage of 24 V. In Figure 14, the amount of ripple content in the output voltage and current is shown, and it can be seen from the simulation that both the output voltage and current have extensively minimal ripple content. Convertor performance parameters are given in Table 4.

Figures 15 and 16 shows the plot for the variations in load voltage and load current when the load changes for the

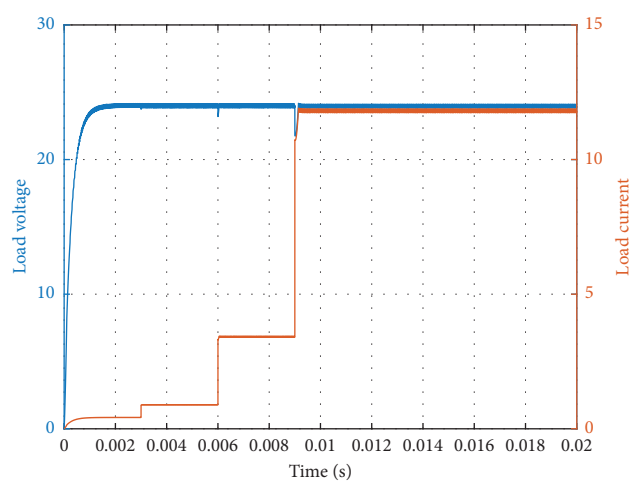

Figure 13: Plot for load current and voltage for varying load using the flower pollination algorithm-tuned QFT controller.

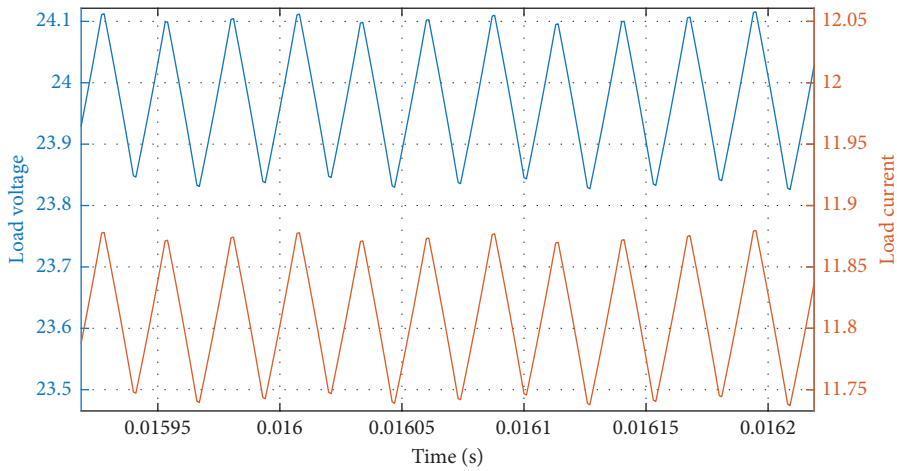

Figure 14: Plot for output current and voltage ripple for varying load case using flower pollination algorithm-tuned QFT controller.

Table 4: Convertor performance parameters (variable load).

Parameters	Voltage ripple (ΔV) (%)	Current ripple (ΔI) (%)
BA	1.458	1.271
FPA	1.0417	1.0294
BBO	1.146	1.102
HS	1.146	1.101
DE	1.25	1.203
ICA	1.146	1.101
IWO	1.25	1.203
TLBO	1.25	1.186
ACO	1.146	1.102
Ibarra	1.36	1.416
IMC	0.724	0.692
ZN	0.267	0.294

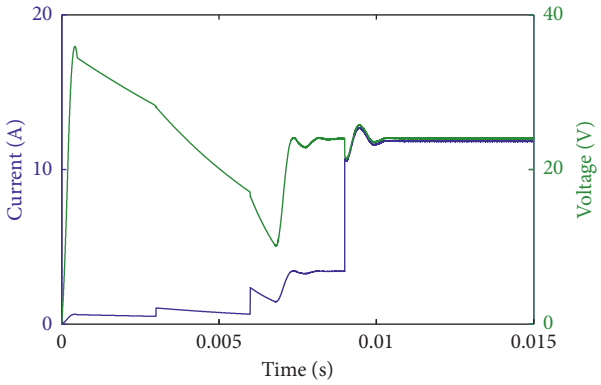

Figure 15: Plot for load current and voltage for varying load using Ziegler–Nichols-tuned PID controller.

Ziegler–Nichols-tuned PID controller and IMC-based controller. From the figures, it can be seen that both these controller fails to stabilize the output voltage when load changes.

Figure 17 shows the plot for the variations in load voltage and load current when the load changes for the QFT controller designed by Ibarra et al. [5].

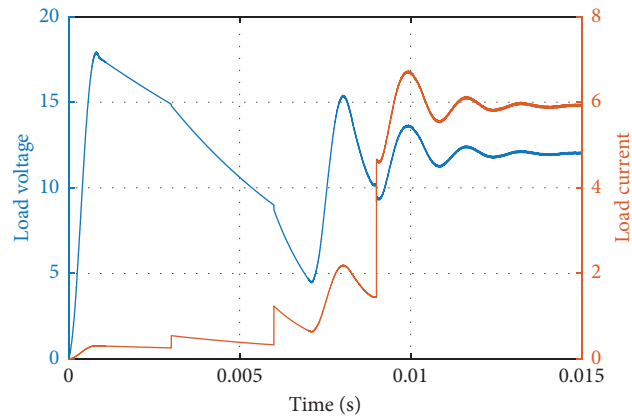

Figure 16: Plot for load current and voltage for varying load using the IMC-based controller.

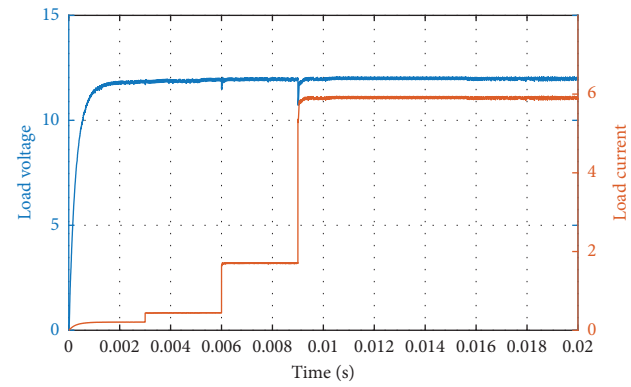

Figure 17: Plot load current and load voltage for varying load using the QFT controller proposed by L Ibarra.

9. Conclusions

In DC-DC convertors, nonlinear behavior due to switching operations and parametric uncertainties due to continuous operations make it hard to yield quality output overtime. In this paper, the automatic synthesis of the QFT controller

and prefilter has been carried out using metaheuristic algorithms. The design process has been posed as an optimization problem, which eliminates the need of generation of templates and bounds and eases the design process. The flower pollination algorithm-based designed controller satisfies the design requirement both in the time and frequency domains and offers better performance than other algorithms. At the end, the designed controller has been implemented for a Simulink model of the DC-DC converter for two different cases of varying input voltage. The designed controller significantly reduces the voltage and current ripples and thus offering a quality voltage and current characteristics.

Data Availability

The data used to support the findings of this study are included within the article.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

References

[1] M. B. Poodeh, S. Eshtehardiha, A. Kiyoumarsi, and M. Ataei, "Optimizing LQR and pole placement to control buck converter by genetic algorithm," in *Proceedings of International Conference on Control, Automation and Systems*, pp. 2195–2200, Seoul, South Korea, October 2007.

[2] A. J. Calderón, B. M. Vinagre, and V. Feliu, "Fractional order control strategies for power electronic buck converters," *Signal Processing*, vol. 86, no. 10, pp. 2803–2819, 2006.

[3] R. D. Middlebrook and S. Cuk, "A general unified approach to modelling switching converter power stages," in *Proceedings of IEEE Power Electronics Specialists Conference*, pp. 18–34, Cleveland, OH, USA, June 1976.

[4] I. Horowitz, *Synthesis of Feedback Systems*, Academic Press, New York, NY, USA, 1963.

[5] L. Ibarra, I. Macıas, P. Ponce, and A. Molina, "On DC/DC voltage buck converter control improvement through the QFT approach," in *Proceedings of International Confrence on New Developments in Circuits, Systems, Signal Processing, Communications and Computers*, pp. 183–190, Vienna, Austria, February 2015.

[6] S. P. Bhattacharyya, H. Chapellat, and L. H. Keel, *Robust Control: the Parametric Approach*, Bergen, NJ, USA, 1995.

[7] C. Li, Z. Ye, Y. Wang, and L. Liu, "Design of attitude decoupling control system for BTT missile using quantitative feedback theory," in *Proceedings of International Conference on Modelling, Identification and Control (ICMIC)*, pp. 819–824, Okayama City, Japan, July 2010.

[8] Z. Shi, F. Wang, and X. Z. Wang, "Design of missile lateral channel controller based on QFT," in *Proceedings of International Conference on Computational and Information Sciences*, pp. 917–920, Los Alamitos, CA, USA, October 2011.

[9] A. R. Saxena and M. Veerachary, "QFT based robust controller design for fourth-order boost dc-dc switching power converter," in *Proceedings of IEEE Joint International Conference on Power Electronics, Drives and Energy Systems (PEDES-2010) Power India*, pp. 1–6, New Delhi, India, December 2010.

[10] A. Khodabakhshian and R. Hemmati, "Robust decentralized multi-machine power system stabilizer design using quantitative feedback theory," *International Journal of Electrical Power and Energy Systems*, vol. 41, no. 1, pp. 112–119, 2012.

[11] S. M. Alavi and M. Saif, "A QFT-based decentralized design approach for integrated fault detection and control," *IEEE Transactions on Control Systems Technology*, vol. 20, no. 5, pp. 1366–1375, 2012.

[12] D. Xu, K. He, Y. Wei, and J. Zhao, "Research on QFT controller design for LOS stabilization system of opto-electronic load for UAV," in *Proceedings of 10th World Congress onIntelligent Control and Automation (WCICA)*, pp. 1982–1986, Beijing, China, July 2012.

[13] L. Ke, W. Zhengzhong, and C. Qiyou, "Design of flight control system for tiltrotor conversion using QFT," in *Proceedings of 26th Chinese Control and Decision Conference (2014 CCDC)*, pp. 3073–3076, Changsha, China, May 2014.

[14] F. Amini and J. Katebi, "Application of a robust QFT linear control in civil engineering," in *Proceedings of 14th World Conference on Earthquake Engineering*, Beijing, China, October 2008.

[15] A. Gera and I. Horowitz, "Optimization of the loop transfer function," *International Journal of Control*, vol. 31, no. 2, pp. 389–398, 1980.

[16] D. J. Ballance and P. J. Gawthrop, "Control systems design via a quantitative feedback theory approach," in *Proceedings of International Conference on Control'91*, pp. 476–480, Edinburgh, UK, March 1991.

[17] R. Nandakumar, G. D. Halikias, and A. Zolotas, "Robust control design of a hydraulic actuator using the QFT method," in *Proceedings of European Control Conference (ECC)*, pp. 2908–2915, Kos, Greece, July 2007.

[18] R. Comasólivas, T. Escobet, and J. Quevedo, "Automatic loop shaping of QFT applied to an active control design," in *Proceedings of 19th Mediterranean Conference on Control and Automation (MED)*, pp. 718–723, Corfu, Greece, June 2011.

[19] O. Yaniv, "Automatic loop shaping of MIMO controllers satisfying sensitivity specifications," *Journal of Dynamic Systems, Measurement, and Control*, vol. 128, no. 2, pp. 463–471, 2006.

[20] A. C. Zolotas and G. D. Halikias, "Optimal design of PID controllers using the QFT method," *Proceedings of Control Theory and Applications*, vol. 146, no. 6, pp. 585–589, 1999.

[21] M. D. Patil, P. S. V. Nataraj, and V. A. Vyawahare, "Automated design of fractional PI QFT controller using interval constraint satisfaction technique (ICST)," *Nonlinear Dynamics*, vol. 69, no. 3, pp. 1405–1422, 2012.

[22] R. Kalla and P. S. V. Nataraj, "Synthesis of fractional-order QFT controllers using interval constraint satisfaction technique," in *Proceedings of 4th IFAC Workshop on Fractional Differentiation and Its Applications (FDA'10)*, Badajoz, Spain, October 2010.

[23] M. D. Patil and P. S. V. Nataraj, "Automated synthesis of multivariable QFT controller using interval constraint satisfaction technique," *Journal of Process Control*, vol. 22, no. 4, pp. 751–765, 2012.

[24] R. Jeyasenthil and P. S. V. Nataraj, "Automatic loop shaping in QFT using hybrid optimization and consistency technique," *IFAC Proceedings Volumes*, vol. 46, no. 32, pp. 427–432, 2013.

[25] M. D. Patil and P. S. V. Nataraj, "Design of robust QFT controllers and prefilters for 3 × 3 distillation column," in *Proceedings of Annual IEEE India Conference (INDICON)*, pp. 1–6, New Delhi, India, December 2013.

[26] P. S. V. Nataraj and M. D. Patil, "Robust control design for nonlinear magnetic levitation system using quantitative feedback theory (QFT)," in *Proceedings of Annual IEEE India Conference (INDICON 2008)*, pp. 365–370, Kanpur, India, December 2008.

[27] A. Goldsztejn, F. Goualard, L. Granvilliers et al., "Robust controller and pre-filter design using QFT and interval constraint techniques," in *Proceedings of 7th International Workshop on Constraint Programming and Decision Making*, Montevideo, Uruguay, October 2014.

[28] M. Garcia-Sanz and J. C. Guillen, "Automatic loop shaping of QFT controllers via genetic algorithm, robust control design (ROCOND 2000)," *IFAC Proceedings Volumes*, vol. 33, no. 14, pp. 603–608, 2000.

[29] W. H. Chen, D. J. Ballance, W. Feng, and Y. Li, "Genetic algorithm enabled computer-automated design of QFT control systems," in *Proceedings of 1999 IEEE International Symposium on Computer Aided Control System Design*, pp. 492–497, Kohala Coast, HI, USA, August 1999.

[30] M. S. Kim and C. S. Chung, "Automatic loop-shaping of QFT controllers using GAs and evolutionary computation," in *Proceedings of AI 2005: Advances in Artificial Intelligence*, pp. 1096–1100, Berlin Germany, May 2005.

[31] C. Molins and M. Garcia-Sanz, "Automatic loop-shaping of QFT robust controllers," in *Proceedings of IEEE 2009 National Aerospace and Electronics Conference (NAECON)*, pp. 103–110, Dayton, OH, USA, July 2009.

[32] B. Satpati, C. Koley, and S. Datta, "Robust PID controller design using particle swarm optimization-enabled automated quantitative feedback theory approach for a first-order lag system with minimal dead time," *Systems Science and Control Engineering: An Open Access Journal*, vol. 2, no. 1, pp. 502–511, 2014.

[33] L. Meng and D. Xue, "Automatic loop shaping in fractional-order QFT controllers using particle swarm optimization," in *Proceedings of IEEE International Conference on Control and Automation ICCA 2009*, pp. 2182–2187, Christchurch, New Zealand, December 2009.

[34] L. Meng and D. Xue, "Design of a new fractional-order QFT controller based on automatic loop shaping," in *Proceedings of 8th World Congress on Intelligent Control and Automation (WCICA)*, pp. 3722–3727, Jinan, China, July 2010.

[35] L. Meng and D. Xue, "QFT fractional order robust controller for non-minimum phase hydro power plant," in *Proceedings of IEEE International Conference on Computer Science and Automation Engineering (CSAE)*, pp. 94–98, Zhangjiajie, China, May 2012.

[36] C. Olalla, R. Leyva, A. El Aroudi, and P. Garcés, "QFT robust control of current-mode converters: application to power conditioning regulators," *International Journal of Electronics*, vol. 96, no. 5, pp. 503–520, 2009.

[37] A. Khodabakhshian, M. Mahdianpoor, and R. A. Hooshmand, "Robust control design for multi-functional DVR implementation in distribution systems using quantitative feedback theory," *Electric Power Systems Research*, vol. 97, pp. 116–125, 2013.

[38] D. Igrec, A. Sarjaš, and A. Chowdhury, "QFT-based robust velocity controller design for a SW-DC motor," *Przegląd Elektrotechniczny*, vol. 87, no. 3, pp. 81–84, 2011.

[39] P. Omer, J. Kumar, and B. S. Surjan, "Design of robust PID controller for buck converter using bat algorithm," in *Proceedings of IEEE 1st International Conference on Power Electronics, Intelligent Control and Energy Systems (ICPEICES)*, pp. 1–5, Delhi, India, July 2016.

[40] N. Katal and S. Narayan, "Optimal QFT controller and pre-filter for buck converter using multi-objective genetic algorithm," *International Journal of Swarm Intelligence*, vol. 3, no. 2-3, pp. 192–214, 2017.

[41] X.-S. Yang, *A New Metaheuristic Bat-Inspired Algorithm*, Springer, Berlin, Germany, 2010.

[42] X.-S. Yang, "Flower pollination algorithm for global optimization," in *Proceedings of UCNC*, pp. 240–249, Orléans, France, September 2012.

[43] D. Karaboga and B. Basturk, "A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm," *Journal of Global Optimization*, vol. 39, no. 3, pp. 459–471, 2007.

[44] D. Simon, "Biogeography-based optimization," *IEEE Transactions on Evolutionary Computation*, vol. 12, no. 6, pp. 702–713, 2008.

[45] Z. W. Geem, J. H. Kim, and G. V. Loganathan, "A new heuristic optimization algorithm: harmony search," *Simulation*, vol. 76, no. 2, pp. 60–68, 2001.

[46] R. Storn and K. Price, "Differential evolution–a simple and efficient heuristic for global optimization over continuous spaces," *Journal of Global Optimization*, vol. 11, no. 4, pp. 341–359, 1997.

[47] B. Xing and W.-J. Gao, "Invasive weed optimization algorithm," in *Innovative Computational Intelligence: A Rough Guide to 134 Clever Algorithms*, Springer, International Publishing, Berlin, Germany, 2014.

[48] R. V. Rao, V. J. Savsani, and D. P. Vakharia, "Teaching–learning-based optimization: a novel method for constrained mechanical design optimization problems," *Computer-Aided Design*, vol. 43, no. 3, pp. 303–315, 2011.

[49] M. Dorigo, M. Birattari, and T. Stutzle, "Ant colony optimization," *IEEE Computational Intelligence Magazine*, vol. 1, no. 4, pp. 28–39, 2006.

[50] H. I. Ali, B. B. Samsul, M. Noor, S. M. Bashi, and M. H. Marhaban, "Quantitative feedback theory control design using particle swarm optimization method," *Transactions of the Institute of Measurement and Control*, vol. 34, no. 4, pp. 463–476, 2011.