Scientific Research Publishing

Journal of Power and Energy Engineering, 2025, 13(1), 47-69
https://www.scirp.org/journal/jpee
ISSN Online: 2327-5901
ISSN Print: 2327-588X

Optimizing Pole Placement Strategies for a Higher-Order DC-DC Buck Converter: A Comprehensive Evaluation

Richard Tymerski

Department of Electrical & Computer Engineering, Portland State University, Portland, Oregon, USA
Email: tymerski@ee.pdx.edu

How to cite this paper: Tymerski, R. (2025) Optimizing Pole Placement Strategies for a Higher-Order DC-DC Buck Converter: A Comprehensive Evaluation. *Journal of Power and Energy Engineering*, **13**, 47-69.
https://doi.org/10.4236/jpee.2025.131004

Received: December 20, 2024
Accepted: January 23, 2025
Published: January 26, 2025

Abstract

This paper explores pole placement techniques for the 4th order C1 DC-to-DC Buck converter focusing on optimizing various performance metrics. Refinements were made to existing ITAE (Integral of Time-weighted Absolute Error) polynomials. Additionally, metrics such as IAE (Integral Absolute Error), ISE (Integral of Square Error), ITSE (Integral of Time Squared Error), a Max-Min metric as well as LQR (Linear Quadratic Regulator) were evaluated. PSO (Particle Swarm Optimization) was employed for metric optimization. Time domain response to a step disturbance input was evaluated. The design which optimized the ISE metric proved to be the best performing, followed by IAE and MaxMin (with equivalent results) and then LQR.

Keywords

DC-DC converter, Integral Time Absolute Error (ITAE), Integral Absolute Error (IAE), Integral Square Error (ISE), Integral Time Squared Error (ITSE), Linear Quadratic Regulator (LQR), Particle Swarm Optimization (PSO)

1. Introduction

Feedback control of DC-DC converters is a critical area of power electronics that has seen significant research and development over the past several decades. As switching power supplies have become ubiquitous in modern electronic systems, there has been an ongoing effort to improve their dynamic performance, efficiency, and reliability through advanced control techniques. While significant progress has been made, there remains a need for comprehensive comparisons of different pole placement and optimization approaches for higher-order converter

models. This paper aims to address that gap by evaluating multiple pole placement methods for a 4th order Buck converter, specifically the C1 converter [1]-[4], under full state feedback with integral control.

The design of controllers based on minimizing specific performance metrics has previously been undertaken for many different converter typologies. Some examples are: the standard 2nd order Buck converter [5]-[9]; Boost converter [10]; Buck-Boost converter [11]; Cuk converter [12]; SEPIC converter [13] and ZETA converter [14]. All four standard performance indices have been examined. These are ITAE (Integral of Time-weighted Absolute Error), IAE (Integral of Absolute Error), ISE (Integral of Squared Error) and ITSE (Integral of Time-weighted Squared Error). The most frequent index cited is ITAE. Perhaps the main reason for this is that prototype polynomial tables exist, [15] and [16], that facilitate design. Newer performance indices also have been introduced such as the ISTAE ('Integral of squared time multiplied by absolute error') [17] and ISTE ('Integral Square Time Error') [6], which refer to the same performance index. For voltage regulators we consider the minimization of the maximum deviation in the output as paramount and so we examine the use of the MaxMin index which simply takes the difference between the maximum and minimum error values.

A plethora of the designs undertaken look specifically at optimizing a PID (three-term) controller or a subset of it. In this work, we consider full state feedback design with integral control. This was previously done for the Buck-Boost converter, a second order system, in [11]. Our 4th order converter will reveal interesting pole placement results. Most recent work has used population based stochastic algorithms to undertake the search. These include: Genetic algorithm (GA) [5]; particle swarm optimization (PSO) [14]; firefly algorithm (FA) [13] and Artificial Bee Colony Algorithm (ABC) [9].

The remainder of this paper is organized as follows: Section 2 presents the mathematical and block diagram models of the converter system. This includes full state feedback (of 4 converter state variables) together with integral control (one added state variable). Section 3 describes a traditional pole placement technique that uses prototype ITAE polynomials. This is examined and refined. Section 4 generalizes the pole placement method and details the optimization approach using a stochastic search algorithm and the various performance criteria of MaxMin, ITAE, IAE, ISE, ITSE and LQR. Simulation results and analysis are also provided in Section 4 where a performance ranking of the optimized indices is presented for the C1 converter. The conclusion follows in Section 5.

2. System Modelling

The C1 PWM converter, shown in **Figure 1**, is a fourth-order DC-to-DC Buck-type converter. This converter has a voltage conversion ratio given by $\frac{V_o}{V_g} = D$ where, V_o, V_g and D are the output voltage, input voltage and duty ratio, respectively. Originally presented in [1] [2] and [3], a more recent

analysis and design methodology has been presented in [4]. An important feature of this converter is the possibility of achieving a minimum phase response which affords an effective feedback design with a wide loop bandwidth. The control design presented in [4] was a classical control approach. As with such approaches, it focused on a transfer function description of the system and developed an effective compensator through loop shaping. In this paper, an alternative modern control design approach is presented and evaluated. A state-space model is developed and full state feedback incorporating integral control is utilized to perform pole placement. A traditional approach of pole assignment using the ITAE (Integral of Time-weighted Absolute Error) metric is first examined. Optimizations of this approach are presented and then expanded upon. Improvements in the traditional ITAE design are presented and other performance metrics are considered, such as IAE (Integral of Absolute Error), ISE (Integral of Squared Error) and ITSE (Integral of Time-weighted Squared Error) and MaxMin (formally defined below). Also, an LQR (Linear Quadratic Regulator) design approach is presented. A population-based stochastic search algorithm is used to optimize the designs.

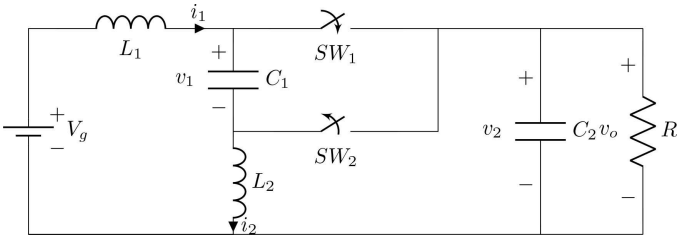

Figure 1. C1 PWM voltage converter.

Under normal operation, the C1 converter cycles through two different circuit configurations, as shown in **Figure 2**, depending on the states of the two switches. During the subinterval of length DT_s, where D is the duty ratio and $T_s = \frac{1}{f_s}$ where f_s is the switching frequency, switches SW_1 and SW_2 are ON and OFF, respectively. Conversely, their positions are reversed during the remainder of the period, of length $(1-D)T_s$.

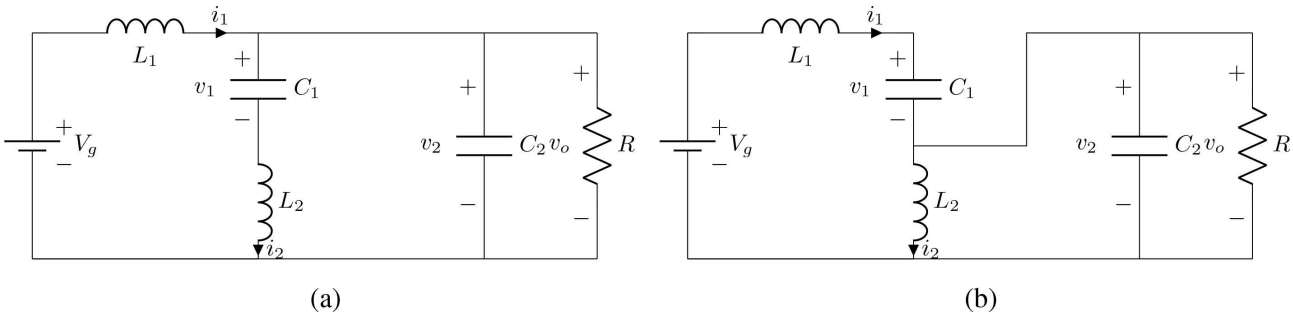

Figure 2. C1 converter circuit configurations: (a) SW_1 is ON, SW_2 is OFF, (b) SW_1 is OFF, SW_2 is ON.

2.1. The State Space Averaged Model

State space averaging [18] is used to model switching converters by calculating a weighted average of the two sets of state equations using the nominal values of the time spent in each state as the weights. A number of key modelling assumptions underlie the validity of the use of this modelling method. One of these is the small ripple approximation, which assumes that variations in inductor currents and capacitor voltages (caused by switching) are small relative to their respective average values. This allows the use of time-averaged values of currents and voltages to represent the dynamics, rather than dealing with high-frequency switching components. A second important assumption is the use of a high switching frequency is assumed to be much higher than the natural frequencies of the converter. This allows the high-frequency switching behavior to be averaged out over a switching period, resulting in a low-frequency equivalent model.

We will define the state vector as follows:

$$x = \begin{bmatrix} v_2 & v_1 & i_2 & i_1 \end{bmatrix}'$$

which are the capacitor voltages and inductor currents as represented in **Figure 1** and **Figure 2**.

$$A_1 = \begin{bmatrix} -\frac{1}{RC_2} & 0 & -\frac{1}{C_2} & \frac{1}{C_2} \\ 0 & 0 & \frac{1}{C_1} & 0 \\ \frac{1}{L_2} & -\frac{1}{L_2} & 0 & 0 \\ -\frac{1}{L_1} & 0 & 0 & 0 \end{bmatrix}, \quad B_1 = \begin{bmatrix} 0 \\ 0 \\ 0 \\ \frac{1}{L_1} \end{bmatrix}, \quad C_1 = \begin{bmatrix} 1 & 0 & 0 & 0 \end{bmatrix} \tag{1}$$

Conversely, when switch SW_2 is ON, with switch SW_1 OFF, the circuit state matrices are given by:

$$A_2 = \begin{bmatrix} -\frac{1}{RC_2} & 0 & -\frac{1}{C_2} & \frac{1}{C_2} \\ 0 & 0 & 0 & \frac{1}{C_1} \\ \frac{1}{L_2} & 0 & 0 & 0 \\ -\frac{1}{L_1} & -\frac{1}{L_1} & 0 & 0 \end{bmatrix}, \quad B_2 = \begin{bmatrix} 0 \\ 0 \\ 0 \\ \frac{1}{L_1} \end{bmatrix}, \quad C_2 = \begin{bmatrix} 1 & 0 & 0 & 0 \end{bmatrix} \tag{2}$$

By representing the steady-state value of the duty cycle of switch SW_1 as D, the following results may be obtained for a state space averaged model of the C1 converter:

$$D' \triangleq 1 - D$$

$$\dot{X} = AX + BV_g$$

$$Y = CX$$

where the averaged matrices are given by:

$$A = DA_1 + D'A_2$$

$$B = DB_1 + D'B_2$$

$$C = DC_1 + D'C_2$$

With the understanding that in steady state $\dot{X} = 0$, we can solve for the steady state vector:

$$X = -A^{-1}BV_g$$

For the C1 converter this vector is determined to be:

$$X = \begin{bmatrix} V_2 \\ V_1 \\ I_2 \\ I_1 \end{bmatrix} = \begin{bmatrix} DV_g \\ V_g \\ \dfrac{DV_g}{R} \\ \dfrac{D^2V_g}{R} \end{bmatrix}$$

Of particular importance here is that the output voltage, which we will designate as V_o is the output capacitor voltage, *i.e.* $V_o = V_2$, indicating that the voltage conversion ratio is given by:

$$\frac{V_o}{V_g} = D \tag{3}$$

Subsequently the small-signal model can be derived. The caret symbol (^) indicates a small deviation from the steady state value, *i.e.*

$$x = X + \hat{x}$$

$$v_g = V_g + \hat{v}_g$$

$$d = D + \hat{d}$$

$$v_o = V_o + \hat{v}_o$$

With the condition that the output matrices $C_1 = C_2$, which appears for the C1 converter, the small signal model is given by:

$$\dot{\hat{x}} = A\hat{x} + B\hat{v}_g + B_d\hat{d} \tag{4}$$

$$\hat{v}_o = C\hat{x} \tag{5}$$

where

$$B_d = (A_1 - A_2)X + (B_1 - B_2)V_g \tag{6}$$

Equations (4) and (5) lead to the general small-signal model block diagram shown in Figure 3.

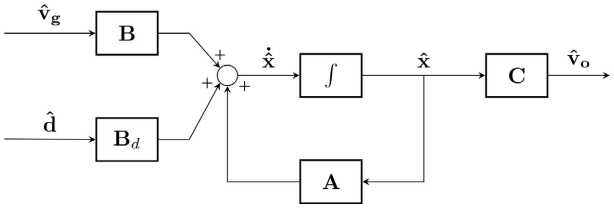

Figure 3. Small-signal block diagram of the open loop system.

2.2. Open Loop Performance

We will now examine the open loop small-signal performance of the converter. To this end a unit step signal will be applied to the $\hat{v}_g$ input and the output response of $\hat{v}_o$ will be examined. As we are examining the open loop response, no control signal will be applied to the $\hat{d}$ input, so that we have $\hat{d}=0$. The converter parameters used are the same as in the design presented in [4]. Specifically, these are summarized in Table 1.

Table 1. Parameter values used in the prototype design.

Parameter	Value
Nominal input voltage, V_g	10 V
Output voltage, V_o	5 V
Steady state duty ratio, D	0.5
Load resistance, R	5 Ω
Switching frequency, f_s	100 kHz
Inductor, L_1	330 μH
Inductor, L_2	680 μH
Capacitor, C_1	10 μF
Capacitor, C_2	10 μF

Note that the model used is a small signal model so any signals that appear will deviate around a zero value. To put this in the context of the actual system a DC voltage of 5 V is added to the output signal. Also, note that a unit step of the input voltage, $\hat{v}_g=1$, represents a 10% change of the nominal input voltage of 10 V. The resulting response is shown in Figure 4.

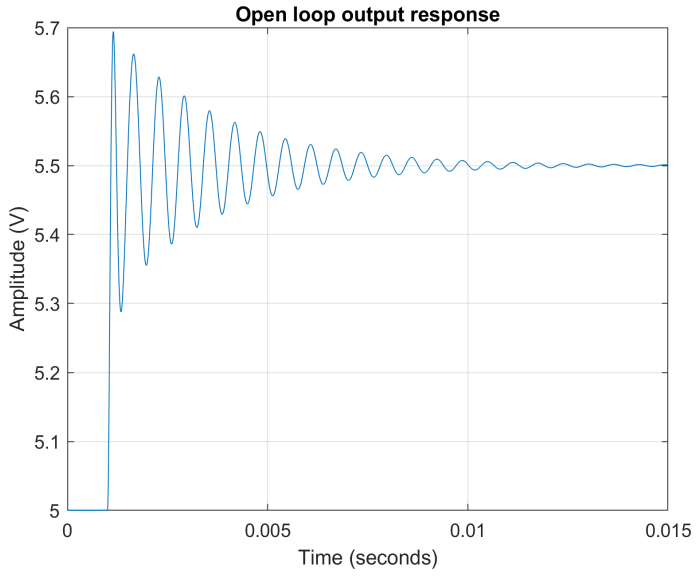

Figure 4. Open loop output response, v_o, to an input unit step, $\hat{v}_g=1$.

In order to clearly see the initial step response, which would be obscured by the vertical axis when the input is applied at $t=0$, the step is applied slightly delayed at $t=1$ ms. Prior to the step's application, the output voltage is at a 5 V level. Subsequently, a peak oscillation value of 5.7 V is reached before settling down to a 5.5 V level, as dictated by the voltage conversion ratio: $V_o = D \cdot \left(V_g + \hat{v}_g\right)$.

For a perfect voltage regulator system, when given a step input voltage disturbance we would like the output to remain unchanged. However, this is not achievable practically and a non-zero step response will be seen in the output. With the incorporation of integral control, which guarantees zero steady state error, the output will revert to its set value in the steady state. The rate, and the manner, that this reversion is achieved will be dictated, but the overall feedback loop design and the speed of response will be primarily dictated by the loop bandwidth. Consequently, measures such as settling time, percentage overshoot, and unity gain bandwidth will be of interest.

In the next section, the basic converter model is augmented to achieve full state feedback with integral control.

2.3. Full State Feedback with Integral Control

The state of a system is defined as a minimal set of variables which together with the input, permits any output to be obtained in a finite time span (if the system is fully controllable). An integrator will also be added to achieve zero steady state error. This will increase the state dimension by one. Our basic converter model will now be modified by adding full state feedback along with integral control. This is shown in Figure 5. The cross (X) at input $\hat{d}$ indicates where the feedback loop may be broken to determine the loop gain of the system.

We will now derive the state equations for this system. As before, for the basic converter we have:

$$\dot{\hat{x}} = A\hat{x} + B\hat{v}_g + B_d\hat{d} \tag{7}$$

The output of the added integrator is the added state, denoted as $\hat{x}_i$, so that at its input we have

$$\dot{\hat{x}}_i = -C\hat{x} \tag{8}$$

These two expressions can be combined into one state equation

$$\begin{bmatrix} \dot{\hat{x}} \\ \dot{\hat{x}}_i \end{bmatrix} = \begin{bmatrix} A & 0 \\ -C & 0 \end{bmatrix} \begin{bmatrix} \hat{x} \\ \hat{x}_i \end{bmatrix} + \begin{bmatrix} B \\ 0 \end{bmatrix} \hat{v}_g + \begin{bmatrix} B_d \\ 0 \end{bmatrix} \hat{d} \tag{9}$$

Converter and integral state feedback is achieved with

$$\hat{d} = -\begin{bmatrix} k & k_i \end{bmatrix} \begin{bmatrix} \hat{x} \\ \hat{x}_i \end{bmatrix} \tag{10}$$

In summary, the above equations can be represented as follows:

$$\dot{\bar{x}} = \bar{A}\bar{x} + \bar{B}\hat{v}_g + \bar{B}_d\hat{d} \tag{11}$$

$$\hat{v}_o = \bar{C}\bar{x} \tag{12}$$

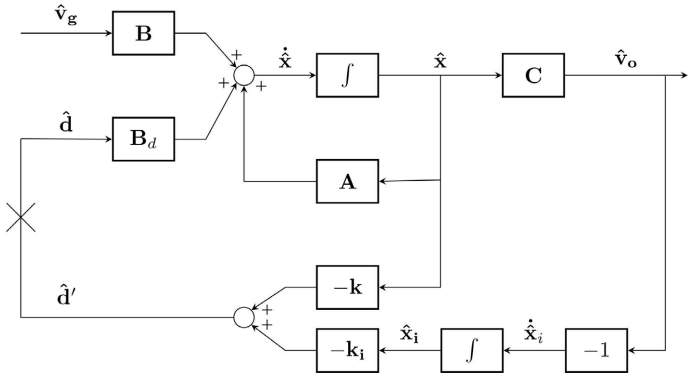

Figure 5. System controlled with a FSFBI regulator.

where the augmented state, $\bar{x}$, is given by

$$\bar{x} = \begin{bmatrix} \hat{x} \\ \hat{x}_i \end{bmatrix} \tag{13}$$

and

$$\bar{A} = \begin{bmatrix} A & 0 \\ -C & 0 \end{bmatrix}, \quad \bar{B} = \begin{bmatrix} B \\ 0 \end{bmatrix}, \quad \bar{B}_d = \begin{bmatrix} B_d \\ 0 \end{bmatrix}, \quad \bar{C} = \begin{bmatrix} C & 0 \end{bmatrix} \tag{14}$$

Also, equation (10) becomes

$$\hat{d} = -\bar{k}\bar{x} \tag{15}$$

where

$$\bar{k} = \begin{bmatrix} k & k_i \end{bmatrix} \tag{16}$$

Substituting (15) into (11) results in the closed loop system state equations:

$$\dot{\bar{x}} = \left(\bar{A} - \bar{B}_d\bar{k}\right)\bar{x} + \bar{B}\hat{v}_g \tag{17}$$

$$\hat{v}_o = \bar{C}\bar{x} \tag{18}$$

The eigenvalues of matrix $\bar{A} - \bar{B}_d\bar{k}$ represent the poles of the closed loop system. These poles can be placed appropriately by varying the feedback gain vector $\bar{k}$. What follows are the various approaches by which this could be done optimally. For the C1 converter there are a total of five feedback gains to be optimized.

As mentioned previously, the cross (X) between the signals $\hat{d}$ and $\hat{d}'$ in **Figure 5** represents the point in the loop which is broken in order to measure the loop gain. With the loop broken at that point, the input signal $\hat{d}$ is fed into the system, (with $\hat{v}_g = 0$), and the output is $\hat{d}'$. The corresponding state equations are

$$\dot{\bar{x}} = \bar{A}\bar{x} + \bar{B}_d\hat{d} \tag{19}$$

$$\hat{d}' = -\bar{k}\bar{x} \tag{20}$$

where the $\bar{A}$, $\bar{B}_d$ and $\bar{k}$ matrices are given in Equations (14) and (16). This results in a loop gain expression given by

$$G_{loop\_gain} = -\bar{k}\left(sI - \bar{A}\right)^{-1}\bar{B}_d \tag{21}$$

In this work we will primarily be looking at the output signal $\hat{v}_o$ in response to a disturbance step input $\hat{v}_g$. State Equations (17) and (18) will be utilized to achieve this. We will also examine the control effort in closed loop where the input signal $\hat{d}$ is monitored. The state equations used for this purpose are therefore (17) and (15).

2.4. Pole Placement and Controllability

As previously stated, the vector of gains $\bar{k}$ determine the poles of the closed loop system. In the work presented here the poles are initially assigned and the gains $\bar{k}$ are subsequently determined. The Matlab functions *acker* or *place* may be used for this purpose. However, for pole placement to be achievable the system must first be determined to be fully controllable. Controllability is defined as the ability to move the state of a system from an initial value x_0 to any arbitrary state x_f within a finite time period t using the input signal u. (Note that controllability says nothing about the magnitude of the input signal u, *i.e.*, the control effort, nor the time t required to accomplish this transition). Essentially, it is a test to determine if the closed-loop system poles may be arbitrarily placed in the complex plane.

A test for controllability can be conducted by first constructing the controllability matrix M_c using the system and input matrices, (A, B) as follows:

$$M_c = \begin{bmatrix} B & AB & A^2B & \cdots & A^{n-1}B \end{bmatrix} \tag{22}$$

where n is the order of the system. If M_c is a full rank matrix, the system is fully controllable. Matlab provides the function *ctrb* to obtain the controllability matrix. The rank deficiency of M_c indicates how many modes are uncontrollable. Using the couple $(\bar{A}, \bar{B}_d)$, as defined in (14), our system was determined to be fully controllable.

3. Traditional Pole Placement Methods

3.1. Pole Placement Methodology

In this paper, several pole placement methods will be explored and compared. To form a basis for comparison, we first examine the traditional and widely accepted ITAE prototype pole placement method.

3.1.1. ITAE Pole Placement Method

Graham and Lathrop [15] discuss assigning the system poles to prototype locations that minimize a performance metric (or cost function), J, known as the integral of the time-weighted absolute error (ITAE) to an input signal:

$$J = \int_0^\infty t \cdot |e(t)| \mathrm{d}t \tag{23}$$

For a system of order n the characteristic polynomial has form $s^n + a_{n-1}a^{n-1} + \cdots + a_2 s^2 + a_1 s + 1$. Note here that the coefficients of the highest and lowest order terms, *i.e.*, the terms involving s^n and s^0, are 1. To minimize the ITAE performance metric, all other coefficients are optimized. Table 2 shows the

traditional values used for these coefficients and corresponding pole positions, p_0, corresponding to a cutoff frequency $\omega_0 = 1$ [15] and [16]. New pole positions, p_N, are then obtained by simple scaling with a different value of ω_0: $p_N = \omega_0 \cdot p_0$.

Table 2. Original ITAE polynomials and pole locations. When ω_0 is not 1, multiple the poles in the table by ω_0. Table adapted from [15] and [16].

Order	Polynomials for $\omega_0 = 1$ rad/s	Poles for $\omega_0 = 1$ rad/s
2	$s^2 + 1.40s + 1$	$-0.7000 \pm 0.7141j$
3	$s^3 + 1.75s^2 + 2.15s + 1$	$-0.5210 \pm 1.0681j$, -0.7081
4	$s^4 + 2.10s^3 + 3.40s^2 + 2.70s + 1$	$-0.4240 \pm 1.2630j$, $-0.6260 \pm 0.4141j$
5	$s^5 + 2.80s^4 + 5.00s^3 + 5.50s^2 + 3.40s + 1$	$-0.3764 \pm 1.2920j$, $-0.8955, -0.5758 \pm 0.5339j$
6	$s^6 + 3.25s^5 + 6.60s^4 + 8.60s^3 + 7.45s^2 + 3.95s + 1$	$-0.3099 \pm 1.2634j$, $-0.5805 \pm 0.7828j$, $-0.7346 \pm 0.2873j$
7	$s^7 + 4.475s^6 + 10.420s^5 + 15.080s^4 + 15.540s^3 + 10.640s^2 + 4.580s + 1$	$-1.2320 \pm 0.9580j$, $-0.2434 \pm 1.0787j$, -0.6860, $-0.4190 \pm 0.5602j$

The ITAE pole placement method sets the system pole locations such that this integral is minimized for a given system cutoff frequency ω_0. The traditional normalized pole locations, which are easily scaled for different ω_0, are given in **Table 2** for up to 7th order systems.

More recent work [19] has highlighted the fact that the original ITAE polynomials and poles were inaccurately determined using an analog computer and an effort to more precisely determine these was made. To illustrate how the updated polynomials are determined, we'll consider the 5th order case, which we'll use later on. The general form of a 5th order polynomial is given as $s^5 + a_4 a^4 + a_3 a^3 + a_2 s^2 + a_1 s + 1$. Thus a search is required to determine the four coefficients a_4, a_3, a_2 and a_1 which minimize the ITAE metric as defined in (23). Use is made of the trapezoidal method of integral approximation: given a time series $t_1, t_2, \cdots, t_n$ and corresponding errors $e_1, e_2, \cdots, e_n$, the ITAE can be approximated as:

$$\text{ITAE} \approx \sum_{i=1}^{n-1} \frac{t_{i+1} - t_i}{2} \left(t_i |e_i| + t_{i+1} |e_{i+1}| \right) \tag{24}$$

Use of the Matlab *trapz* function is made to perform the integration. If we consider the system response to a unit step, the error signal e, is defined as $e(t) = 1 - y(t)$ where $y(t)$ is the system output. Appendix A shows a Matlab script, incorporating the ideas discussed here, which can be used to determine the updated coefficients for polynomials of order 2 to 7. The updated set of ITAE polynomials and pole locations are summarized in **Table 3**. The values displayed in

Table 3 are slightly different from those appearing in [19], due to an unfortunate error in coding used to produce the values given in [19].

Table 3. Updated ITAE polynomials and pole locations. When ω_0 is not 1, multiply the poles in the table by ω_0. The polynomials seen here were obtained by the running the Matlab code given in **Appendix A**.

Order	Updated polynomials for $\omega_0 = 1$ rad/s	Updated poles for $\omega_0 = 1$ rad/s
2	$s^2 + 1.5049s + 1$	$-0.7525 \pm 0.6586j$
3	$s^3 + 1.7828s^2 + 2.1715s + 1$	$-0.5368 \pm 1.0592j$, -0.7092
4	$s^4 + 1.9521s^3 + 3.3458s^2 + 2.6473s + 1$	$-0.3893 \pm 1.3253j$, $-0.5868 \pm 0.4240j$
5	$s^5 + 2.0668s^4 + 4.4976s^3 + 4.6731s^2 + 3.2568s + 1$	$-0.2897 \pm 1.5086j$, $-0.4637 \pm 0.7360j$, -0.5600
6	$s^6 + 2.1519s^5 + 5.6290s^4 + 6.9338s^3 + 6.7925s^2 + 3.7398s + 1$	$-0.2203 \pm 1.6398j$, $-0.3662 \pm 0.9742j$, $-0.4894 \pm 0.3125j$
7	$s^7 + 2.2169s^6 + 6.7433s^5 + 9.3469s^4 + 11.5770s^3 + 8.6778s^2 + 4.3226s + 1$	$-0.1701 \pm 1.7381j$, $-0.2920 \pm 1.1578j$, $-0.4101 \pm 0.5642j$, -0.4726

3.1.2. ITAE Initial Results

As mentioned above, the system discussed in this paper is 5th order, so traditionally to make use of the ITAE pole placement method, an appropriate ω_0 value needs first to be identified to scale the five poles given in the tables. For our purposes, this was done by simply sweeping ω_0 through a range of values, evaluating the step response of the system, and identifying the ω_0 that resulted in the smallest output variation. The sweep interval used was [800, 100,000] rad/s. The left plot of Figure 6 shows the value of the ITAE metric over this interval. The lowest value of ITAE of the order of $2e^{-8}$ appears at the upper ω_0 boundary of 100,000 rad/s. A simulation using this value of ω_0 to determine the output response to a unit step, is shown in the right plot of Figure 6. Unfortunately, a peak to peak variation of 30 volts is seen in the output. This is untenable for a practical design. It appears that the ITAE metric has been minimized by increasing the speed of response without consideration of output voltage deviations.

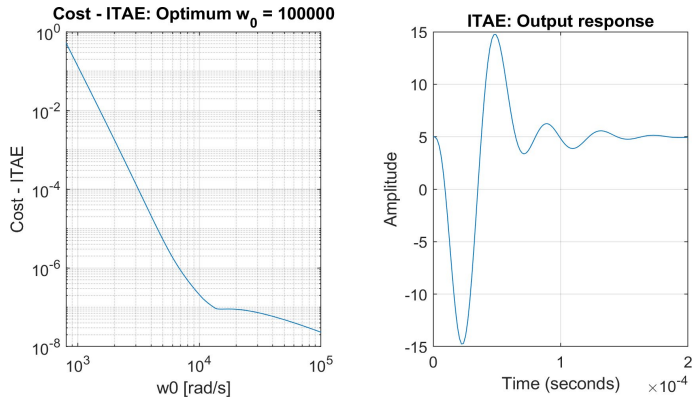

Figure 6. Choosing ω_0 to minimize ITAE: minimum ITAE occurs at $\omega_0 = 100000$, see left plot. However, for this value of ω_0, the output response to a unit step input shows a 30 V variation, see right plot.

In consideration of the above results, we next decided to monitor the MaxMin metric defined by

$$\text{MaxMin} = e_{\max}(t) - e_{\min}(t) \tag{25}$$

Accordingly, over the ω_0 sweep range where the ITAE system polynomial is used for design, the MaxMin metric is evaluated, as seen in the left-side plot of Figure 7. Here we find that in order to achieve minimum output variation, as defined by Equation (25), a value of $\omega_0 = 11640$ rad/s is needed. The resulting output plot for this value of ω_0, for a unit step input, is shown in the right-side plot of Figure 7. The peak to peak output amplitude variation has been reduced to around 1.2 volts. Whilst this is a big improvement from the former case, it is also not acceptable.

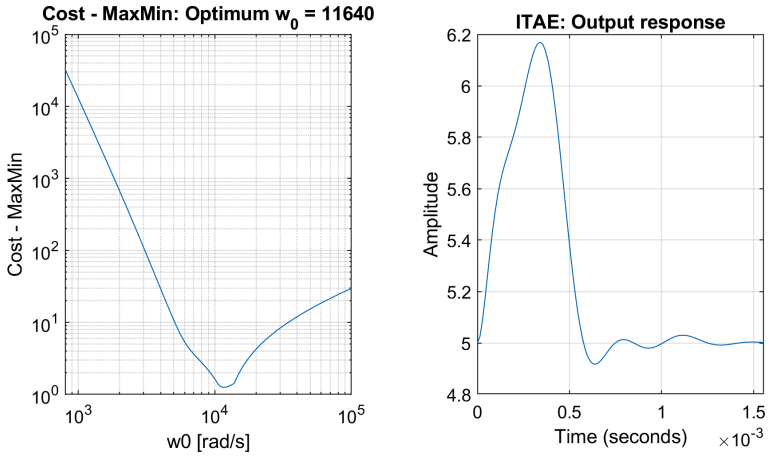

Figure 7. Choosing ω_0 to minimize the MaxMin metric of the ITAE based design: Minimum MaxMin occurs at $\omega_0 = 11640$, as seen in the left plot. However, this results in a 1.2 V range for the output response to a unit step input, as seen in the right plot.

4. Generalized Approach using Various Performance Metrics

From the above, we can see that the standard ITAE pole placement approach is not well suited for the C1 converter. Consequently, in the following we consider for our 5th order system a characteristic polynomial of the form $s^5 + a_4 s^4 + a_3 s^3 + a_2 s^2 + a_1 s + a_0$ where contrary to the previous approach where $a_0 = 1$ was initially assigned, (which corresponded to a cutoff frequency $\omega_0 = 1$), we will now optimize all coefficients, a_4, a_3, a_2, a_1 and a_0, simultaneously to minimize the cost function. The constraint on bandwidth will be accomplished by limiting the pole positions. After some initial trial and error, the bounds of the real and imaginary components of any pole of ≤30,000 were established. In other words, the poles, $s_i, i = 1, \cdots, 5$, will be confined to be within, or on the boundary of, a rectangle in the s-plane with bounds on the right and left given by the imaginary axis (*i.e.*, the line $s_i = 0$) and the line $s_i = -30000$ and on the top and bottom by the horizontal lines $s_i = 30000\,j$ and $s_i = -30000\,j$, respectively.

Furthermore, we examined the performance of a number of different cost func-

tions. The six cost functions considered are listed in Table 4. The definitions and Matlab implementations are given there. Where integration is needed, the use of trapezoidal numerical integration is made with the Matlab *trapz* function. All four standard cost functions are examined: ITAE, IAE (Integral of Absolute Error), ISE (Integral of Squared Error) and ITSE (Integral of Time weighted Squared Error). Since we are dealing with a voltage regulator control system, the metric of the difference between maximum and minimum error deviations was considered sufficiently important, leading to the MaxMin cost function being defined as seen in the table and Equation (25). The last cost function considered is that of the LQR (Linear Quadratic Regulator) problem. A quick review of the LQR problem and solution is provided in **Appendix B**.

In contrast to the other methods, in the LQR case, pole positions are not directly assigned, but rather through manipulating the weighting matrices, Q and R. For our case of a single control input, R is a scalar which was set to $R=1$. The matrix Q is a 5th order diagonal matrix such that $Q=\mathrm{diag}\left(q_1,q_2,q_3,q_4,q_5\right)$. Our search algorithm will vary the values of $q_i, i=1,\cdots,5$ used in the solution of the LQR problem. Note that if a valid solution is found, then due to the properties of LQR design, there are accompanying guaranteed performance attributes, such as a phase margin > 60˚ and an infinite gain margin. To obtain a solution to the LQR problem, matrix Q needs to be positive semi-definite. For a diagonal matrix, this requires that the diagonal values $q_i \geq 0$. After some trial simulation runs, the search bounds on $q_i, i=1,\cdots,5$ were set to $1\mathrm{e}^{-6} \leq q_i, i=1,\cdots,5 \leq 1\mathrm{e}^{8}$, which avails considerable leeway in assigning these weights. A search algorithm is used together with the LQR solution to optimize the following two results: 1) minimize the MaxMin metric as defined in (25), and 2) achieve a desired unity gain crossover frequency, f_c. The frequency chosen was one close to that achieved by the other methods so as to facilitate performance comparison between these methods. This frequency was set to $f_c = 12400$ Hz.

The search algorithm used to assign pole locations, as used in the top 5 cost functions of Table 4, and weights in the Q matrix of the LQR problem is Particle Swarm Optimization (PSO) [20], which is a stochastic search algorithm. A short description of this algorithm is provided in Appendix C. In our implementation, use of the Matlab *particleswarm* function is made. The relevant code is shown next:

```
nvars = 5;                      % number of variables to be adjusted
lb = 1e-6*ones(1,nvars);        % lower bound of exploration range
ub = 1e8*ones(1,nvars);         % upper bound of exploration range

options = optimoptions('particleswarm','HybridFcn',@fmincon);
q = particleswarm(func,nvars,lb,ub,options);

R = 1;                          % input weighting scalar
Q = diag(q);                    % optimized state weights
kbar = lqr(Abar,Bdbar,Q,R);     % kbar is the solution to the LQR problem
```

Note that use of the "'*HybridFcn*',*@fmincon*" option is made as this algorithm

further refines the solution after *particleswarm* terminates. The function *func*, which appears as a parameter in the *particleswarm* function, is where the MaxMin metric and the unity gain crossover frequency are determined. The PSO search endeavors to simultaneously minimize MaxMin and the difference between the observed unity gain crossover frequency and the desired crossover frequency.

Table 4. Optimization metrics with cost function definitions and MATLAB implementations.

Metric	Cost function	MATLAB implementation
MaxMin	$e_{\max}(t)-e_{\min}(t)$	max(e) - min(e)
ITAE	$\int_0^\tau t\lvert e(t)\rvert \mathrm{d}t$	trapz(t, t.*abs(e))
IAE	$\int_0^\tau \lvert e(t)\rvert \mathrm{d}t$	trapz(t, abs(e))
ISE	$\int_0^\tau e^2(t)\mathrm{d}t$	trapz(t, e.^2)
ITSE	$\int_0^\tau te^2(t)\mathrm{d}t$	trapz(t, t.*e.^2)
LQR	$\int_0^\infty \left(x^{\mathrm{T}}Qx+u^{\mathrm{T}}Ru\right)\mathrm{d}t$	k = lqr(A, B, Q, R) Finds k, subject to Q and R, such that $u=-kx$ minimizes the cost function.

Results

Performance of the various methods is assessed in the time domain by examining the step response and, in the frequency domain, by examining the loop gain Bode plot. Step responses are shown in Figure 8 and Bode plots are shown in Figure 9. In the time domain plots, the control effort signal is also shown. For our system, the control effort is the duty ratio, d. The duty ratio needs to be well constrained within the interval $0<d<1$. The steady state duty ratio is $D=0.5$ $\left(=\frac{V_o}{V_g}\right)$, so deviations from this level are examined and must be within the interval stated.

Relevant results for all methods are summarized in Table 5. The performance metrics examined and appearing in Table 5 are 1) ST (μs): the 0.2% settling time. A 0.2% value was chosen as the overshoot was quite small. This figure determines the length of time for the output to converge to a band of ±0.01 V around the 5 V steady state output, *i.e.*, 4.99 V to 5.01 V. 2) OS (%): the percentage overshoot. This is based on a final settling value of 5 V, 3) Peak (V): the peak overshoot value, 4) f_c (Hz): loop unity gain crossover frequency, 5) PM (˚): loop phase margin in degrees, 6) GM (dB): loop gain margin in decibels, 7) Poles 1 - 5: locations of all 5 pole positions determined by the various methods, 8) k_1 - k_5: the feedback gain vector, where $\bar{k}=\begin{bmatrix}k_1 & k_2 & k_3 & k_4 & k_5\end{bmatrix}$.

Perhaps the first result to be pointed out is that the time domain and frequency domain performance using the MaxMin and IAE metrics were found to be, for intents and purposes, identical to each other. Evidently, examining the peak excursions values of the error, *i.e.*, maximum and minimum, as undertaken to evaluate the MaxMin metric, is sufficiently equivalent to using all the error excursions,

i.e. $|e(t)|$, over the examined interval as used in evaluating the IAE index. Accordingly, results for these appear in Table 5 under the single title *MaxMin & IAE*. However, the step response and loop gain Bode plots for both are shown in Figure 8 and Figure 9 so that readers can confirm this observation for themselves.

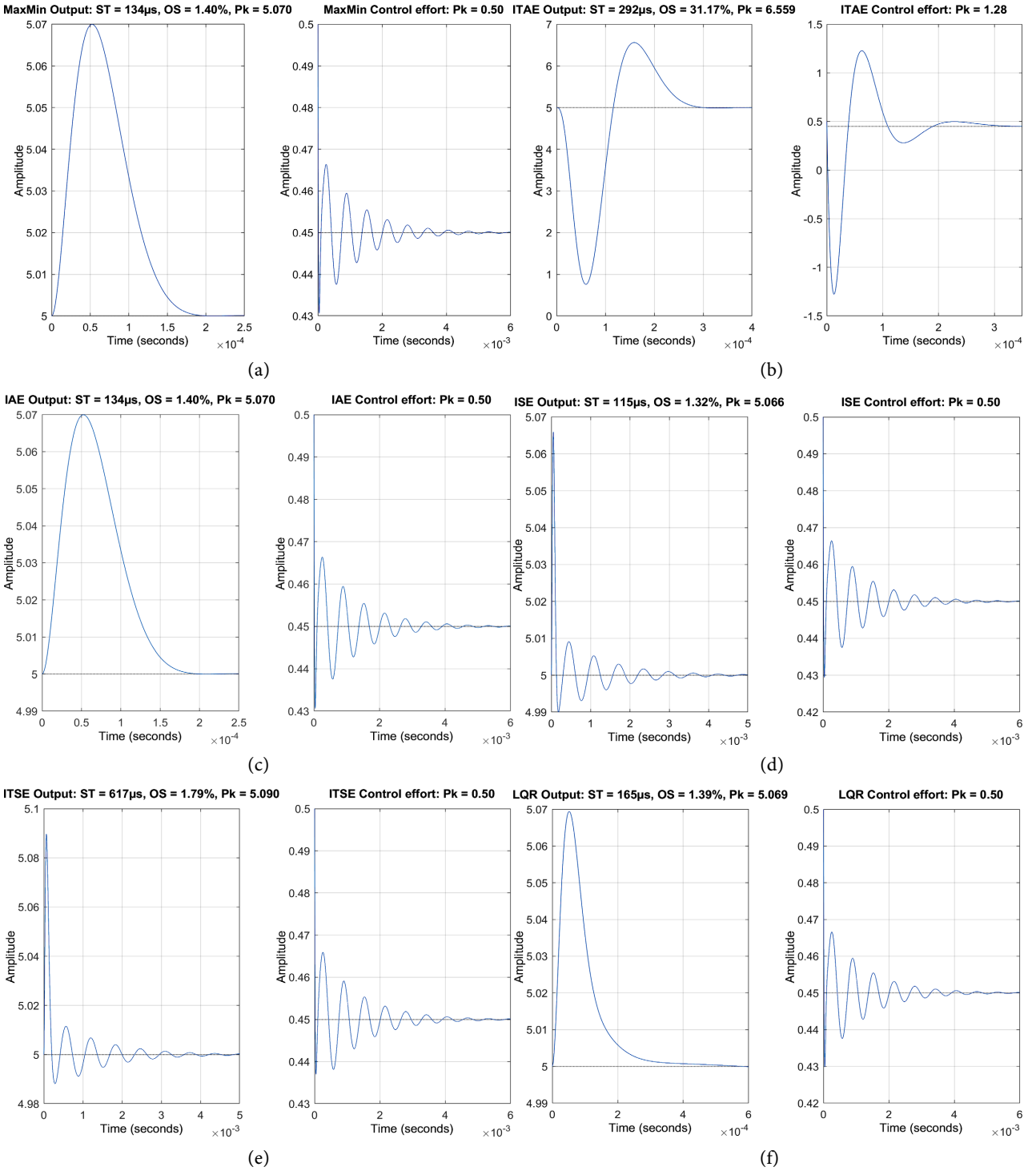

Figure 8. Output v_o (on the left) and control effort, *i.e.*, duty ratio d, (on the right) responses for the different cost functions. (a) MaxMin, (b) ITAE, (c) IAE, d) ISE, (e) ITSE and (f) LQR. Note that (a) and (c) show identical results.

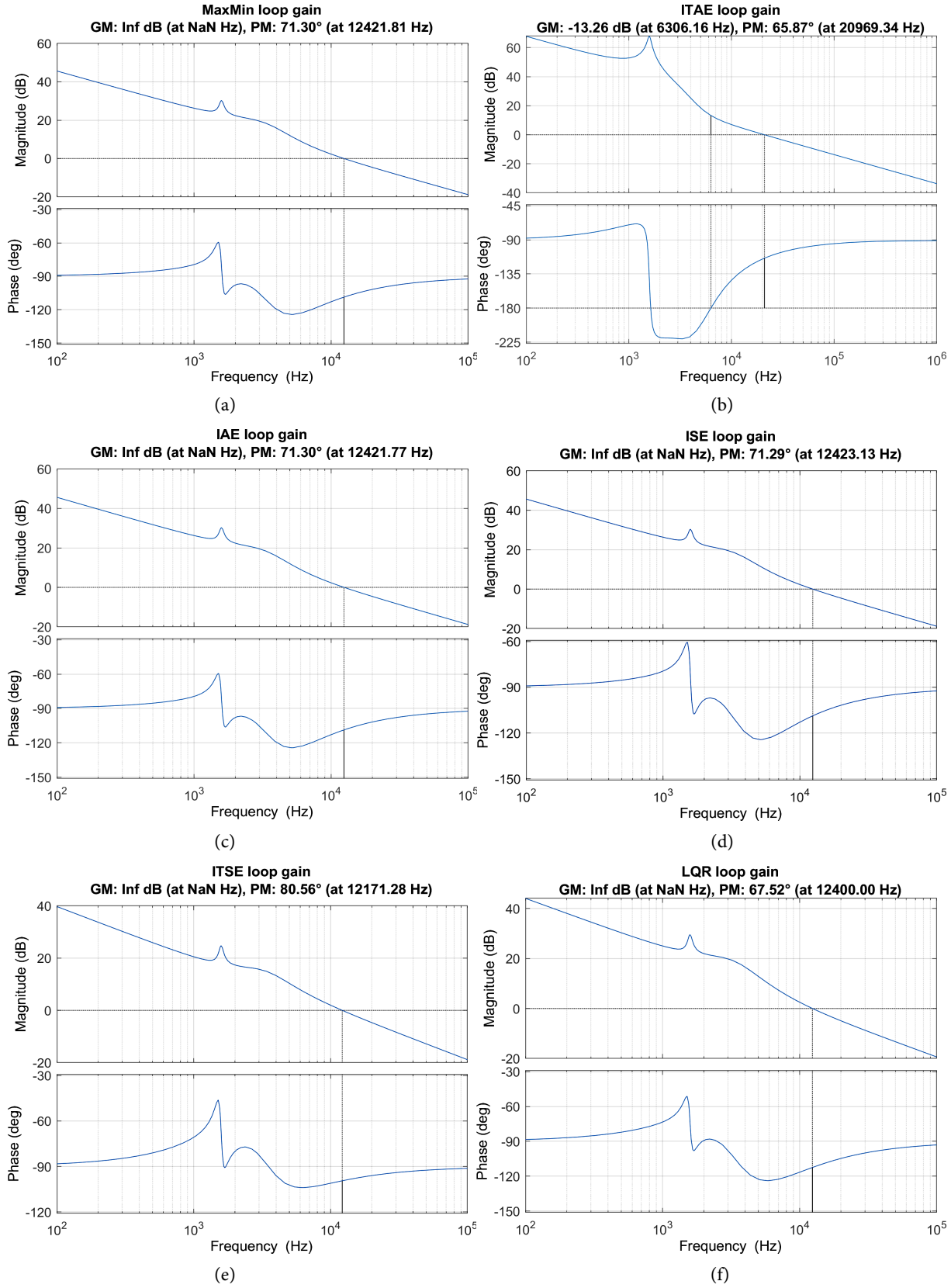

Figure 9. Loop gain Bode plots for the different cost functions. (a) MaxMin, (b) ITAE, (c) IAE, (d) ISE, (e) ITSE and (f) LQR. Note that (a) and (c) show almost identical results, differing only very slightly in the unity gain crossover frequency.

Table 5. Comparison of Control Performance Metrics Across Different Criteria. 1) ST (μs): 0.2% settling time, 2) OS (%): percentage overshoot. This is based on a final settling value of 5V, 3) Peak (V): the peak overshoot value, 4) f_c (Hz): loop unity gain crossover frequency, 5) PM (˚): loop phase margin in degrees, 6) GM (dB): loop gain margin in decibels, 7) Poles 1 - 5: locations of all 5 pole positions determined by the various methods, 8) k_1 - k_5: the feedback gain vector, where $\bar{k} = \begin{bmatrix} k_1 & k_2 & k_3 & k_4 & k_5 \end{bmatrix}$.

	MaxMin & IAE	ITAE	ISE	ITSE	LQR
ST (μs)	133.80	291.89	114.60	616.82	164.52
OS (%)	1.3998	31.171	1.3182	1.7912	1.3882
Peak (V)	5.0700	6.5586	5.0659	5.0896	5.0694
f_c (Hz)	12421.81	20969.34	12423.13	12171.28	12,400
PM (°)	71.30	65.87	71.29	80.56	67.52
GM (dB)	∞	−13.26	∞	∞	∞
Poles 1 & 2	$-30000 \pm 30000j$	$-30000 \pm 30000j$	$-30000 \pm 30000j$	$-30000 \pm 3628.5j$	$-34716 \pm 39727j$
Poles 3 & 4	$-866.34 \pm 9912.6j$	$-30000 \pm 9912.6j$	$-873.62 \pm 9938.6j$	$-837.70 \pm 9936.2j$	$-867.18 \pm 9913.8j$
Pole 5	−30,000	−30,000	−30,000	−29,894	−16,166
k_1	0.38879	−7.4474	0.38562	0.18939	0.47614
k_2	−0.017327	27.499	−0.0043643	−0.0082966	−0.016881
k_3	−1.5552	32.491	−1.5610	−1.6103	−1.4553
k_4	1.5552	110.81	1.5957	1.5530	1.4600
k_5	−11,998	−153,420	−12,062	−6,090.7	−10,000

Before considering method performance, we first examine the achieved pole locations of all methods. An interesting pattern emerges when examining the pole positions as given in Table 5: 1) Examining positions of Poles 1 and 2, for most indices a complex pole pair was placed at the boundary limits of $-30000 \pm 30000j$, or close by in the case of the LQR method where no direct limits were imposed, where the complex pole pair appears at $-34716 \pm 39727j$. The results of the ITSE method were different, with the pole pair appearing at $-30000 \pm 3628.5j$; 2) Similarly, examining Pole 5 position; this real pole was placed at the limit of −30,000, or close by at −29,894 for the ITSE index. However, this was not the case for the LQR method where the pole appears at −16,166, which is closer to the imaginary axis; 3) Examining position of Poles 3 and 4, we see a complex pole pair placed at around $-870 \pm 9.920j$ for all indices except ITAE where the pair appear at $-30000 \pm 9912.6j$, pushing the poles much further away from the imaginary axis. The effects on the output time response will be discussed below. Note that the PSO search in the LQR method resulted in finding the optimized weighting matrix

$Q = \text{diag}(3.6191\text{e}-01, 3.6315\text{e}-04, 2.0445\text{e}-02, 4.0713\text{e}-02, 1.0\text{e}+08)$.

Next, when examining all responses, we see that using the ITAE method is unacceptable, as we had seen previously. The output voltage deviations are extreme, with a high value of 6.559 V and low value of around 0.8 V. The variation in duty ratio (i.e., control effort) is also seen to be beyond practical bounds. Examining

the loop gain Bode plot shows that even though the system is stable with an acceptable phase margin of 65.87˚, it nevertheless features a negative gain margin of −13.26 dB. The time and frequency domain performances of all remaining methods are considered acceptable. We will rank these in the ensuing discussion.

Having eliminated the ITAE method at this point, the following relates to the remaining methods only. The unity gain crossover frequencies f_c are in the range $f_c \in [12171, 12423]$ Hz with phase margins $\mathrm{PM} \in [67.52^\circ, 80.56^\circ]$ and all feature infinite gain margins. The narrow range of crossover frequencies validates a comparison between the methods. In regards to time domain performance, the overshoot values are in a narrow range $\mathrm{OS}(\%) \in [1.3182, 1.7912]$ with a peak overshoot value of around 5.07 V, a deviation of just 70 mV from the 5 volt steady state value. The improvement over the open loop response shown in **Figure 4** is well noted. Perhaps the greatest performance differentiator is the settling time. In this regard, the ISE method is seen to be the best with 114 μs and the least favorable being ITSE with 616.82 μs. MaxMin and IAE gave slightly better results of 133.80 μs compared to that of LQR with 164.52 μs.

In summary, with all the above considered, for our C1 converter system, the ISE pole placement method gave the best results. This is followed up by the MaxMin & IAE methods and LQR method.

An appreciation as to why the ITAE and ITSE methods under-perform can be gained by considering the form of the cost function for each of these metrics. In each case, the integrand features a 't' (*i.e.*, time) multiplier term. The consequence of this is that initial errors, *i.e.*, when the value of t is small, are under-weighted and do not contribute sufficiently to the total metric value. Accordingly, initial large error deviations, such as those that appear for step input disturbances as used here to assess performance, are permitted. The overshoot for the ITAE metric was seen to be 31.171%. This under-weighting of the error term in the metric is somewhat alleviated in the ITSE case by having the error squared.

A close look at the loop plots for the ISE, IAE, and MaxMin indices shown in Figure 9 shows that the performance is identical (for IAE and MaxMin) or near identical (for ISE and IAE & MaxMin). This is expected when noting the pole locations for these methods. The slight edge noted for the ISE index in terms of slightly shorter settling time is a consequence of poles, Poles 3 and 4, (with all others identical between the methods) which are located slightly further away from the imaginary axis.

The LQR method features similar pole locations, for Poles 3 and 4, however, the real pole, Pole 5, appears at a location much closer to the imaginary axis and has a more dominant effect in slowing down the settling time.

5. Conclusions

In this paper, we examined pole placement techniques for the C1 DC-to-DC converter, which optimized various performance metrics. A commonly applied metric is the ITAE due to the availability of a standard set of ITAE polynomials which

may be used for design. However, we discovered that the existing ITAE polynomials were somewhat inaccurate, prompting us to refine them and develop a more precise set. Initially, we applied ITAE in a standard manner for pole placement in controller design. This process typically involves using a polynomial with a cutoff frequency (ω_0) of one, then scaling the poles according to a different chosen cutoff frequency. However, when this approach was applied to the C1 converter, it resulted in excessive overshoot and impractical control effort, *i.e.*, wide impractical swings in the duty ratio.

To expand our investigation, we included additional standard performance metrics such as IAE, ISE, and ITSE. We also looked at a metric we have termed MaxMin, which measures the error by taking the difference between the maximum and minimum error values. We also undertook an LQR design, where the weighting matrices were chosen to minimize the MaxMin metric while also assigning an appropriate loop unity gain crossover frequency. In performing the optimizations needed, we utilized a stochastic population-based search algorithm, specifically Particle Swarm Optimization, which proved to be very effective. Performance analysis included examining step response metrics, such as settling time and overshoot, control effort (measured by duty ratio variation), and loop gain Bode plot measures such as unity gain crossover frequency and phase margin. We ranked the resulting performance of the optimized indices. The optimized ISE index proved to be the best performing overall, as it yielded the least settling time, whereas other performance metrics were similar for competing indices. Following ISE, the MaxMin and IAE designs showed identical strong performance, while the LQR ranked just below them. Notably, ITAE and ITSE are not recommended due to their suboptimal performance in this context.

This study was primarily theoretical and conducted via simulations. For practical implementation, we recommend the use of digital control based on a non-minimum state-space description, as this approach avoids the need for an observer and the associated robustness issues [21]. Future work will focus in this direction.

Conflicts of Interest

The author declares no conflicts of interest regarding the publication of this paper.

References

[1] Tymerski, R. and Vorperian, V. (1986) Generation, Classification and Analysis of Switched-Mode Dc-to-Dc Converters by the Use of Converter Cells. *International Telecommunications Energy Conference*, Toronto, 19-22 October 1986, 181-195. https://doi.org/10.1109/intlec.1986.4794425

[2] Tymerski, R. and Vorperian, V. (1988) Generation and Classification of PWM Dc-To-Dc Converters. *IEEE Transactions on Aerospace and Electronic Systems*, **24**, 743-754. https://doi.org/10.1109/7.18641

[3] Tymerski, R. (1988) Topology and Analysis in Power Conversion and Inversion. PhD Dissertation, Department of Electrical Engineering, Virginia Polytechnic Institute

and State University.

[4] Simmons, J. and Tymerski, R. (2021) Design and Control of an Alternative Buck PWM Dc-to-Dc Converter. *Journal of Power and Energy Engineering*, **9**, 43-61. https://doi.org/10.4236/jpee.2021.96004

[5] Nishat, M.M., Faisal, F., Evan, A.J., Rahaman, M.M., Sifat, M.S. and Rabbi, H.M.F. (2020) Development of Genetic Algorithm (GA) Based Optimized PID Controller for Stability Analysis of DC-DC Buck Converter. *Journal of Power and Energy Engineering*, **8**, 8-19. https://doi.org/10.4236/jpee.2020.89002

[6] Alfergani, A., Elkawafi, S., Nour, T.M., Mohamed Elkezza, K. and Kahlil, A. (2023) Performance Evaluation of DC-DC Buck Converter with Voltage Control Loop Using Genetic Algorithm with Different Objective Functions. 2023 *IEEE* 3*rd International Maghreb Meeting of the Conference on Sciences and Techniques of Automatic Control and Computer Engineering* (*MI-STA*), Benghazi, 21-23 May 2023, 135-140. https://doi.org/10.1109/mi-sta57575.2023.10169481

[7] Asadi, F. and Abut, N. (2016) ITAE Criterion Based Controller for Buck Converter. *International Journal of Advanced and Applied Sciences*, **4**, 15-22. https://doi.org/10.21833/ijaas.2017.01.003

[8] Rodrigues, I.P., Raposo Cardoso Sabino, I., da Silva Câmara, M., Bernardes Alves Machado, M.H. and Guedes Alves, P. (2024) Assessment of IAE and Itae-Based Objective Functions for PID Controller Tuning in Buck Converters within Space Systems. *Cadernos UniFOA*, **19**, 1-10. https://doi.org/10.47385/cadunifoa.v19.n54.4898

[9] Tajudin, A.I., Adika, M.A., Abd Samat, A.A., Abd Shukor, S.F. and Daud, K. (2021) Design an Optimal PI Controller Using Artificial Bee Colony Algorithm for Buck Converter. 2021 6*th IEEE International Conference on Recent Advances and Innovations in Engineering* (*ICRAIE*), Kedah, 1-3 December 2021, 1-7. https://doi.org/10.1109/icraie52900.2021.9703778

[10] Mohajery, R., Shayeghi, H. and Lopez Villanueva, J.A. (2021) PSO-TVAC Based Optimized FOPID Controller for Cascaded DC-DC Boost Converter. *International Journal on Technical and Physical Problems of Engineering*, **13**, 139-146.

[11] Al-Baidhani, H., Sahib, A. and Kazimierczuk, M.K. (2023) State Feedback with Integral Control Circuit Design of DC-DC Buck-Boost Converter. *Mathematics*, **11**, Article No. 2139. https://doi.org/10.3390/math11092139

[12] Thirumeni, M. and Thangavelusamy, D. (2019) Design and Analysis of Hybrid PSO-GSA Tuned PI and SMC Controller for DC-DC Cuk Converter. *IET Circuits, Devices & Systems*, **13**, 374-384. https://doi.org/10.1049/iet-cds.2018.5164

[13] Shagor, M.R.K., Mahmud, A.J., Nishat, M.M., Faisal, F., Mithun, M.H. and Khan, M.A. (2021) Firefly Algorithm Based Optimized PID Controller for Stability Analysis of DC-DC SEPIC Converter. 2021 *IEEE* 12*th Annual Ubiquitous Computing, Electronics & Mobile Communication Conference* (*UEMCON*), New York, 1-4 December 2021, 957-963. https://doi.org/10.1109/uemcon53757.2021.9666555

[14] Nishat, M.M., Shagor, M.R.K., Akter, H., Mim, S.A. and Faisal, F. (2020) An Optimal Design of PID Controller for DC-DC Zeta Converter Using Particle Swarm Optimization. 2020 23*rd International Conference on Computer and Information Technology* (*ICCIT*), Dhaka, 19-21 December 2020, 1-6. https://doi.org/10.1109/iccit51783.2020.9392676

[15] Graham, D. and Lathrop, R.C. (1953) The Synthesis of "Optimum" Transient Response: Criteria and Standard Forms. *Transactions of the American Institute of Electrical Engineers, Part II: Applications and Industry*, **72**, 273-288. https://doi.org/10.1109/tai.1953.6371346

[16] Franklin, G.F., Emami-Naeini, A. and Powell, J.D. (1994) Feedback Control of Dynamic Systems. 3rd Edition, Addison-Wesley Longman Publishing Co., Inc.

[17] Shayeghi, H., Salmalian, S., Mohajery, R. and Bizon, N. (2024) Design and Performance Evaluation of Two-Degree-of-Freedom PID Control for a Modified Boost Converter. 2024 16*th International Conference on Electronics, Computers and Artificial Intelligence* (*ECAI*), Iasi, 27-28 June 2024, 1-6. https://doi.org/10.1109/ecai61503.2024.10606971

[18] Middlebrook, R.D. and Cuk, S. (1976) A General Unified Approach to Modelling Switching-Converter Power Stages. 1976 *IEEE Power Electronics Specialists Conference*, Cleveland, 8-10 June 1976, 18-34. https://doi.org/10.1109/pesc.1976.7072895

[19] Cao, Y. (2020) The Optimal ITAE Transfer Function for Step Input. MATLAB Central File Exchange. https://www.mathworks.com/matlabcentral/fileexchange/18547-the-optimal-itae-transfer-function-for-step-input

[20] Eberhart, R. and Kennedy, J. (1995) A New Optimizer Using Particle Swarm Theory. *MHS'*95. *Proceedings of the* 6*th International Symposium on Micro Machine and Human Science*, Nagoya, 4-6 October 1995, 39-43. https://doi.org/10.1109/mhs.1995.494215

[21] Taylor, C.J., Young, P.C. and Chotai, A. (2013) True Digital Control: Statistical Modeling and Non-Minimal State Space Design. Wiley. https://doi.org/10.1002/9781118535523

Appendix A: Updated ITAE Polynomials: MATLAB Code

The MATLAB code shown below can be used to determine polynomials of order 2 to 7, (with cutoff frequency $\omega_0 = 1$), which minimizes the ITAE metric for a step input. The resulting polynomials (and roots) are given in **Table 3**. Note that these coefficient values are slightly more accurate than those found in [19]. The algorithms starts with the original ITAE polynomial coefficients, as given in **Table 2**, and refines them.

```
clear; format short g

p_original = {1.4, [1.75 2.15], [2.1 3.4 2.7], [2.8 5.0 5.5 3.4], ...
              [3.25 6.60 8.60 7.45 3.95], [4.475 10.42 15.08 15.54 10.64 4.58]};

for order = 2:7
  p0 = p_original{order-1};          % use original ITAE coefficients as initial guess
  dt = 0.0001;                       % use small simulation time steps to improve accuracy
  tfn = 50;                          % use lengthy simulation of step response for accuracy
  cost = @(x)itae_cost(x,dt,tfn);    % directs calls to 'itae_cost' function
  [p, itae] = fminunc(cost,p0);      % given initial guess p0, find p which minimizes ITAE
  p_updated = [1, p, 1]              % updated ITAE polynomial coefficients
  itae                               % ITAE value
end

function itae = itae_cost(p,dt,tfn)
  G = tf(1, [1, p, 1]);              % set up system transfer function
  t = 0:dt:tfn;                      % time vector
  [y, tx] = step(G, t);              % Step response of the system
  e = 1 - y;                         % error signal
  itae = trapz(tx, tx.*abs(e));      % ITAE integral approximation
end
```

Appendix B: LQR Control Problem and Solution

The Linear Quadratic Regulator (LQR) is a method used in optimal control theory to design a state feedback controller that minimizes a *cost function* while stabilizing a linear dynamical system.

The Problem

The system is typically represented by a set of linear state-space equations:

$$\dot{x}(t) = Ax(t) + Bu(t)$$

where $x(t)$ is the state vector, $u(t)$ is the control input, A and B are system matrices.

The objective is to find a control input $u(t) = -Kx(t)$ (state feedback) that minimizes a quadratic *cost function*:

$$J = \int_0^\infty \left(x^{\mathrm{T}} Q x + u^{\mathrm{T}} R u\right) \mathrm{d}t$$

where Q is a positive semi-definite matrix that penalizes deviations in the state, R is a positive definite matrix that penalizes the control effort.

The Solution

The optimal control law is derived by solving the Algebraic Riccati Equation (ARE):

$$A^{\mathrm{T}} P + PA - PBR^{-1} B^{\mathrm{T}} P + Q = 0$$

where P is the solution to the Riccati equation.

Once P is obtained, the optimal gain matrix K is calculated as:

$$K = R^{-1} B^{\mathrm{T}} P$$

This results in the optimal control law:

$$u(t) = -Kx(t)$$

which minimizes the cost function J and provides the best trade-off between state deviations and control effort.

Appendix C: Particle Swarm Optimization

Particle Swarm Optimization (PSO) is an iterative computational technique used to identify an optimal solution for a specified criterion in a given search space by having a population of solution sets, referred to as "particles", moving through the search space in a set algorithm. This technique makes use of the memory of each particle's past movement and optimal solution set found, and the knowledge of the best solution found by all particles to move throughout the search space in such a way that it increases the probability of the particles converging to a single, optimal solution. The fact that the movement of each particle is based on the memory of the individual, in addition to the experiences of the search group as a whole, emulating social patterns seen in nature, is where PSO derives its name from. The actual motion of the particles themselves is controlled by their velocity, which is randomly chosen, and their "flying" towards a global optima. The equations that dictate this motion are given by:

$$v_i^{k+1} = w \cdot v_i^k + c_1 \cdot r_1 \left(p_i^k - x_i^k \right) + c_2 \cdot r_2 \left(g^k - x_i^k \right)$$

$$x_i^{k+1} = x_i^k + v_i^{k+1}$$

where v is the velocity, w is the inertial constant, c and r are positive constants used to dictate the importance of the previous velocity and that of the personal and global bests, g is the global best, p is the personal best, x is the particle position, i is the particle, and k is the iteration index. For the purposes of this paper, w was set to a linearly varying value ranging from 0.9 to 0.4 over the course of the PSO iterations, while the c and r values were randomly initialized with respect to their initial weights.