mathematics

MDPI

Article

Reinforcement Learning-Based Control of a Power Electronic Converter

Dajr Alfred [1,*,†], Dariusz Czarkowski [1,†] and Jiaxin Teng [2,†]

[1] Department of Electrical and Computer Engineering, New York University, 5 MetroTech Center, Brooklyn, NY 11201, USA; dariusz.czarkowski@nyu.edu
[2] Texas Instruments, Sugar Land, TX 77479, USA; jteng@nyu.edu
* Correspondence: dva240@nyu.edu
† These authors contributed equally to this work.

Abstract: This article presents a modern, data-driven, reinforcement learning-based (RL-based), discrete-time control methodology for power electronic converters. Additionally, the key advantages and disadvantages of this novel control method in comparison to classical frequency-domain-derived PID control are examined. One key advantage of this technique is that it obviates the need to derive an accurate system/plant model by utilizing measured data to iteratively solve for an optimal control solution. This optimization algorithm stems from the linear quadratic regulator (LQR) and involves the iterative solution of an algebraic Riccati equation (ARE). Simulation results implemented on a buck converter are provided to verify the effectiveness and examine the limitations of the proposed control strategy. The implementation of a classical Type-III compensator was also simulated to serve as a performance comparison to the proposed controller.

Keywords: model-free; data-driven control; optimal control; iterative ARE algorithm; reinforcement learning-based control; nonlinear gain scheduling

MSC: 93B99

Citation: Alfred, D.; Czarkowski, D.; Teng, J. Reinforcement Learning-Based Control of a Power Electronic Converter. *Mathematics* **2024**, *12*, 671. https://doi.org/10.3390/math12050671

Academic Editor: António Lopes

Received: 18 January 2024
Revised: 16 February 2024
Accepted: 21 February 2024
Published: 25 February 2024

1. Introduction

The effects of climate change have expedited the need for more widespread adoption of green energy solutions. Efforts such as the United Nations Climate Change Conference's (UNCCC) COP27 [1] and COP28 help to raise global awareness and create actionable objectives to combat this pertinent issue. One of the main issues highlighted at the Conference of the Parties (COP) is the continued global over-reliance on fossil fuels. To combat this issue, the use of renewable energy-based alternatives such as solar- and wind-powered distributed energy resources (DERs), electric vehicles (EVs) and energy storage solutions has been presented [2,3]. The introduction of these modern, technological solutions to the aging power grid, however, presents several challenges [4].

Among these issues are problems involving the control and regulation of these new technologies. As an example, consider the fact that renewable energy DERs are direct-current (DC) power sources delivering power to an alternating-current (AC) power grid. Another example is that of the DC load that a charging EV places on the AC power grid. Clearly, devices are required to perform the necessary conversions of power. Power electronic converters are energy conversion devices that convert electrical energy from one form or level to another form/level. For example, an inverter is a DC-AC power electronic device able to convert DC power at one level to AC power at another level, while a rectifier is an AC-DC converter able to convert AC power to DC power. Most power electronic converters rely on high-frequency switching of silicon-based transistors with control being achieved via modulation of the control signal feeding these transistors. Frequency modulation and pulse width modulation (PWM) are popular power electronic control

schemes. Several power electronic converters feature high-frequency (HF) transformers that are able to transmit AC signals from their primary side(s) to their secondary side(s). These HF transformers offer galvanic isolation of the input and output sides of the power electronic converter.

Evidently, power electronic converters are required in both of these cases to allow for the efficient transfer of power from one form and amplitude to another. The need for grid-scale power electronics means that suitable converters are required to feature high power densities and high efficiencies in order to facilitate power transfer at very high power levels. Therefore, complex, high-frequency switching converters are preferred. A suitable class of converter is the resonant power electronic converter class [5]. However, such converters are inherently nonlinear and classical controllers, derived from approximate linear models of nonlinear resonant converters, are only feasible for a narrow operating range. Our paper aims to address this control issue by presenting a novel RL-based optimal controller that does not rely on model dynamics but rather uses measured data along system trajectories to derive a control solution.

Optimal control involves offline minimization of performance functions, based on Hamilton–Jacobi–Bellman (HJB) design equations, with complete knowledge of system dynamics [6]. In contrast, adaptive control involves dynamically learning control solutions using measured data, online, with no prior knowledge of system dynamics. RL-based control aims to combine these two control methodologies. In control systems, RL refers to a family of techniques used to design optimal adaptive controllers with novel structures that learn the solutions to optimal control performance functions in real time by observing data along the system trajectories [7]. As seen in Figure 1, RL techniques often feature an actor–critic structure. The critic evaluates the reward/cost of the current control policy using feedback from the environment/system of the effect of the current control action. The critic evaluates the response from the environment by calculating the cost/reward via a value function. The actor updates/improves the control policy/action and implements the new/improved control policy. Two RL techniques of note are policy iteration and value iteration, which evaluate the performance of current control policies and provide methods for improving those policies [7]. Policy iteration and value iteration use the Bellman equation to solve optimal control problems forward in time. Using value function approximation, these methods can be implemented online using standard adaptive control system identification algorithms such as recursive least squares (RLS) [8].

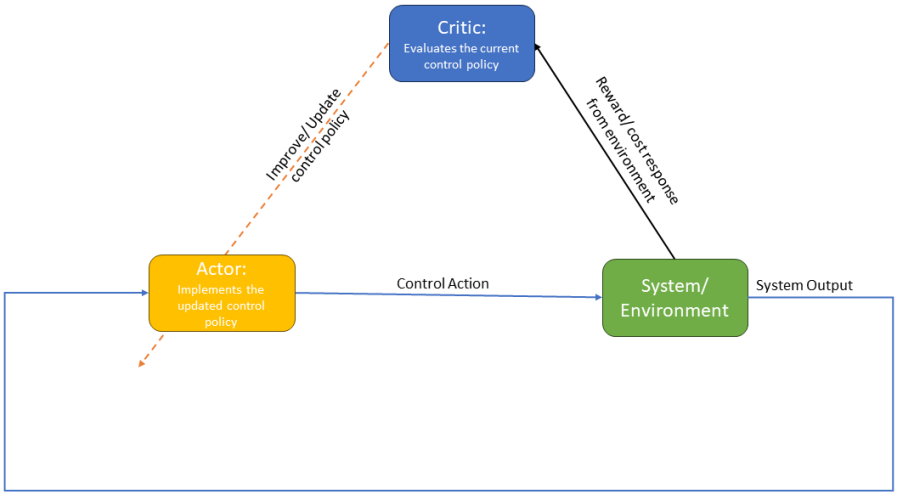

Figure 1. Actor–critic reinforcement learning structure.

The use of reinforcement learning (RL) techniques to obtain adaptive optimal controllers for output regulation of power electronic converters has become a topic of interest over the past several years. Gao and Jiang (2015, 2016) presented an algorithm for adaptive optimal output regulation of linear systems with unknown system dynamics and

immeasurable disturbance [9,10]. These papers focused on continuous-time linear systems but indicated that the algorithms presented were also applicable to nonlinear systems. Ref. [9] provided simulation results on an LCL-coupled inverter-based distributed generation system; hence, it provides evidence of the efficacy of RL control for complex power electronic systems. More recently, a 2023 paper [11] successfully deployed a model-free, deep reinforcement learning (DRL) algorithm on a three-level neutral point clamped (NPC) converter. This paper provides irrefutable evidence of the suitability of RL-based controllers for optimal output regulation of complex, nonlinear power electronic converters.

In their 2019 paper, Jiang et al. presented a data-driven, reinforcement learning-based approach for solving the output regulation problem for discrete-time systems [12]. Their paper collected several key RL concepts and featured detailed analysis of mathematical concepts. Their paper presented three optimal feedback control algorithms. Algorithm 1 solved for an optimal feedback control solution via policy iteration using Hewer's algorithm [7], directly finding the solution to the discrete-time algebraic Riccati equation (DARE) offline with complete knowledge of the system dynamics. Algorithm 2 relaxed the need for knowledge of system dynamics and presented a model-free optimal feedback control approach. Finally, Algorithm 3 presented a data-driven approach for solving the output regulation problem. Their paper focused solely on control theory, but the application of this theory to power electronic converters had not yet been considered. Therefore, using this paper as a foundation, our previous work, [13], utilized Algorithms 2 and 3 from [12] to obtain an optimal output control solution for buck and boost converters. Hence, additional evidence of the efficacy of the proposed RL techniques for control of power electronic converters was presented. This current paper aims to extend our previous work by introducing new application techniques of the proposed RL algorithm, commenting on limitations of the proposed control method while also further expounding on the underlying mathematical concepts.

2. Materials and Methods

2.1. Derivation of the Composite System for Output Regulation

Consider the discrete-time (DT), linear time-invariant (LTI) system shown in Figure 2 with dynamics

$$\begin{aligned} x(k+1) &= Ax(k) + Bu(k) + E_o d(k) \\ y(k) &= Cx(k) + Du(k) + F_o d(k) \end{aligned} \tag{1}$$

and with tracking error

$$e(k) = y(k) - r(k) = Cx(k) + Du(k) + F_o d(k) - r(k), \tag{2}$$

where $x \in \mathbb{R}^{n_x}$ is the state, $u \in \mathbb{R}^{n_u}$ is the input, $d \in \mathbb{R}^{n_d}$ is the disturbance, $r \in \mathbb{R}^{n_r}$ is the reference, $y \in \mathbb{R}^{n_y}$ is the output, and $n_y = n_r$. $A \in \mathbb{R}^{n_x \times n_x}$, $B \in \mathbb{R}^{n_x \times n_u}$, $E_o \in \mathbb{R}^{n_x \times n_d}$, $C \in \mathbb{R}^{n_y \times n_x}$, $D \in \mathbb{R}^{n_y \times n_u}$, and $F_o \in \mathbb{R}^{n_x \times n_d}$ are constant matrices.

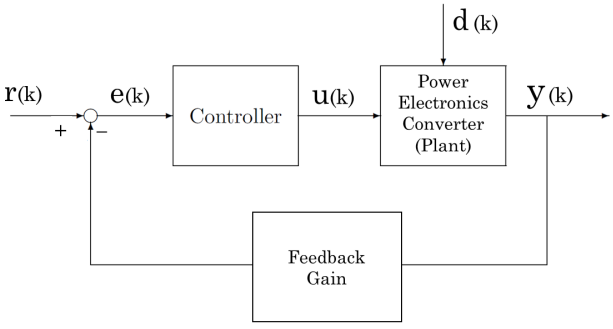

Figure 2. Diagram of closed-loop, discrete-time, linear time-invariant system with a power electronic converter as the controlled plant.

Let us assume that the disturbance and reference signals are generated by dynamics described by

$$\begin{aligned} r_1(k+1) &= M_r r_1(k) \\ r(k) &= G r_1(k) \\ r_1(0) &= r_0, \end{aligned} \tag{3}$$

where $r_1 \in \mathbb{R}^{n_{r_1}}$, $M_r \in \mathbb{R}^{n_{r_1} \times n_{r_1}}$, $G \in \mathbb{R}^{n_r \times n_{r_1}}$ and

$$\begin{aligned} d(k+1) &= M_d d(k) \\ d(0) &= d_0, \end{aligned} \tag{4}$$

where $M_d \in \mathbb{R}^{n_d \times n_d}$. These exosystem dynamics can be augmented to give

$$v(k+1) = \begin{bmatrix} r_1(k+1) \\ d(k+1) \end{bmatrix} = \begin{bmatrix} M_{r_1} & 0 \\ 0 & M_d \end{bmatrix} \begin{bmatrix} r_1(k) \\ d(k) \end{bmatrix} = Mv(k), \tag{5}$$

where $M \in \mathbb{R}^{n_v \times n_v}$, $n_v = n_{r_1} + n_d$.

Finally, the composite system can be formulated by combining Equations (1), (2), and (5) to give

$$\begin{aligned} x(k+1) &= Ax(k) + Bu(k) + Ev(k) \\ v(k+1) &= Mv(k) \\ e(k) &= Cx(k) + Du(k) + Fv(k), \end{aligned} \tag{6}$$

where $E \in \mathbb{R}^{n_x \times n_v} = [0 \quad E_o]$ and $F \in \mathbb{R}^{n_y \times n_v} = [-G \quad F_o]$.

Equation (6) indicates that the system tracking error, $e(k)$, can be considered the output of the composite system. Also, including the exosystem dynamics, namely, the reference and/or disturbance signal dynamics, in the composite model has the benefit of increasing robustness of the closed-loop system to variations in these signals. For composite systems of the form shown in Equation (6), the problem of output, $y(k)$, tracking the reference signal, $r(k)$, as well as the problem of disturbance rejection constitute the output regulation problem.

2.2. The Linear Optimal Output Regulation Problem

The linear optimal output regulation problem (LOORP) refers to the derivation of an optimal control input $u^*(k)$ for the system with open-loop dynamics described by Equation (1) and closed-loop dynamics described by Equation (6) that ensures closed-loop stability and in which the output $y(k)$ asymptotically tracks the reference, $r(k)$ [12]. The tracking problem is equivalent to having the error signal, $e(k)$, asymptotically regulated to zero:

$$\lim_{k\to\infty} e(k) = \lim_{k\to\infty} (y(k) - r(k)) = \lim_{k\to\infty} (Cx(k) + Du(k) + F_o d(k) - Gr_1(k)) = 0. \tag{7}$$

One class of controller suitable for solving the LOORP is the static-state feedback controller of the form

$$u^*(k) = -K_x^* x(k) + K_v^* v(k), \tag{8}$$

where $K_x \in \mathbb{R}^{n_u \times n_x}$ and $K_v \in \mathbb{R}^{n_u \times n_v}$. The feedback gain, K_x, is designed to ensure that $(A - BK_x)$ is Schur, meaning all eigenvalues of $(A - BK_x)$ are inside the unit circle of the z-plane, hence ensuring exponential stability of the closed-loop system. The feedfor-

ward gain, K_v, and unknown constant matrix, X, are designed such that they satisfy the following equations:

$$\begin{aligned} XM &= (A - BK_x)X + BK_v + E \\ 0 &= (C - DK_x)X + DK_v + F, \end{aligned} \tag{9}$$

where we have applied the optimal controller of Equation (8) to the composite system given by Equation (6).

Theorem 1. *The linear output regulation problem is solvable by a static feedback controller of Equation (8) iff there exist two constant matrices K_v and X that solve Equation (9).*

Proof. The proof of this theorem can be found in the proof of Lemma 1.6 in [14]. □

We can use the linear transformation

$$\begin{bmatrix} X \\ U \end{bmatrix} = \begin{bmatrix} I_{n_x} & 0_{n_x \times n_u} \\ -K_x & I_{n_u} \end{bmatrix} \begin{bmatrix} X \\ K_v \end{bmatrix} \tag{10}$$

where $X \in \mathbb{R}^{n_x \times n_v}$ and $U \in \mathbb{R}^{n_u \times n_v}$ to reformulate Equation (9) as

$$\begin{aligned} XM &= AX + BU + E \\ 0 &= CX + DU + F, \end{aligned} \tag{11}$$

which are known as the *regulator equations*. Hence, Theorem 2 can be formulated as:

Theorem 2. *Given that $(A - BK_x)$ is exponentially stable, the linear output regulation problem is solvable by a static feedback controller of Equation (8) iff there exist two constant matrices X and U that solve Equation (11) with K_v given by*

$$K_v = U + K_x X. \tag{12}$$

Proof. This theorem is proven by the proof of Theorem 1 and the linear transform given in Equation (10). □

Theorem 3. *For all matrices E and F, the regulator equations are solvable iff*

$$rank\left(\begin{bmatrix} A - \lambda I & B \\ C & D \end{bmatrix}\right) = n_x + n_v, \forall \lambda \in \sigma(M).$$

Proof. The proof of this theorem is given as the proof of Theorem 1.9 in [14]. □

We shall assume that the requirements of Theorems 2 and 3 are met. We also assume that the system of Equation (1) is both controllable, which allows arbitrary placement of closed-loop poles via feedback gain, K_x, and observable, which allows for full state feedback via the system output, $y(k)$.

Consider, now, the closed-loop system formed by applying the optimal static feedback controller of Equation (8) to the system whose dynamics are described by Equation (6).

By post-multiplying the regulator equations of Equation (11) by $v(k)$, we obtain

$$\begin{aligned} XMv(k) = Xv(k+1) &= AXv(k) + BUv(k) + Ev(k) \\ 0 &= CXv(k) + DUv(k) + Fv(k). \end{aligned} \tag{13}$$

By subtracting Equation (13) from Equation (6) and omitting exosystem dynamics, we obtain

$$\begin{aligned} x(k+1) - Xv(k+1) &= A[x(k) - Xv(k)] + B[u(k) - Uv(k)] \\ e(k) &= C[x(k) - Xv(k)] + D[u(k) - Uv(k)]. \end{aligned} \tag{14}$$

Now, by defining the variables $\overline{x} = x(k) - Xv(k)$ and $\overline{u} = u(k) - Uv(k)$, we obtain the error system dynamics:

$$\begin{aligned} \overline{x}(k+1) &= A\overline{x} + B\overline{u} \\ e(k) &= C\overline{x} + D\overline{u}. \end{aligned} \tag{15}$$

2.3. Development of the RL Framework

2.3.1. MDPs and the Bellman Equation

Figure 3 depicts a Markov decision process (MDP). MDPs are random/stochastic sequences/processes of possible states where the probability of a future state depends solely on the current state of the system and not on other, prior states. MDPs provide a framework for the development of RL techniques [7,8]. We see in Figure 3 that each transition from state x_i to state x_j, and taking input/action u_l, where $i, j \in [1,3]$ and $l \in [1,2]$, is associated with a transition probability, $P^{u_l}_{x_i x_j}$, and transition cost, $R^{u_l}_{x_i x_j}$. The transition probability, $P^{u_l}_{x_i x_j} = Pr\{x_j | x_i, u_l\}$, is the conditional probability of the system transitioning to state x_j, given that the system starts at state x_i and takes action u_l. The transition cost, $R^{u_l}_{x_i x_j} = E\{r_k | x_k = x_i, u_k = u_l, x_{k+1} = x_j\}$, is the expected value of the stage cost, r_k, at time k.

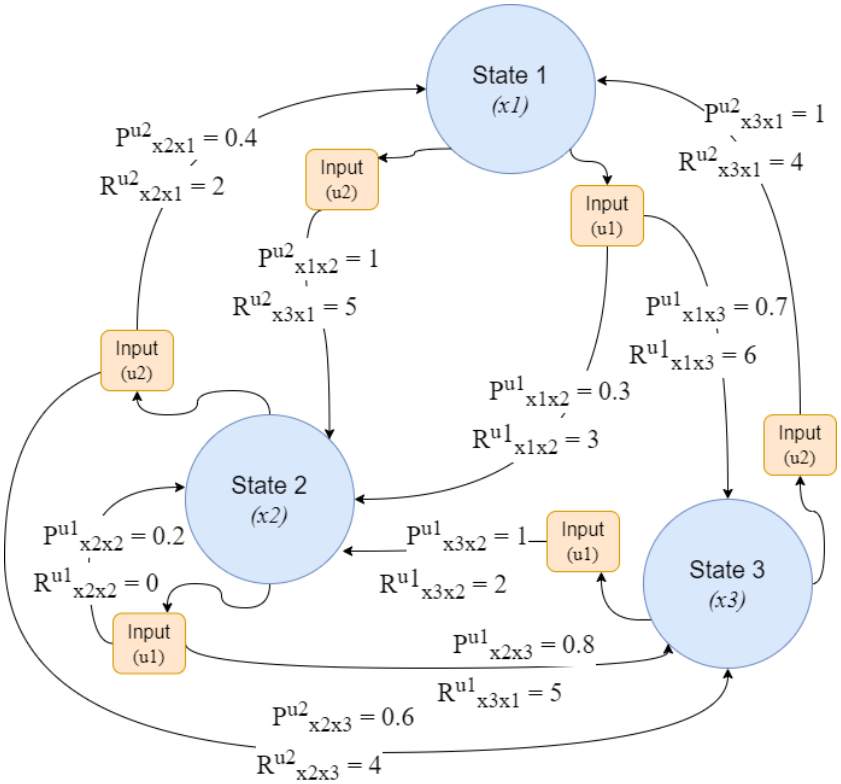

Figure 3. Markov decision process in the form of a finite-state machine with controlled state transitions and costs associated with each transition.

The fundamental MDP problem is to find a mapping that gives the conditional probability $\pi(x,u) = Pr\{u|x\}\ \forall x, u$ of taking action u given that the system/MDP is currently in state x. Such a mapping is termed a closed-loop control/action strategy or policy [7,8].

Consider now the deterministic case of the policy $\pi(x,u)$, where for any given current state, x_i, the only possible action taken by the system is u_i. Hence, $\pi(x,u) = Pr\{u|x\} = \mu(x)$ is a deterministic function, mapping states to inputs. Such a mapping, $\mu(x)$, is a solution to the linear optimal output regulation problem if it maps the states of an LTI system to a stabilizing optimal input that minimizes tracking error. The previous assumptions of controlability and observability of the system described by Equations (1) and (6) allow for such a closed-loop control policy to be derived.

In order to find the optimal control policy, $\pi^*(x,u) = \mu^*(x)$, we must assess the performance of each policy by assigning value/cost to each control policy. Firstly, for each policy, we calculate the discounted sum of future costs over a time period $[k, k+T]$ as

$$J_{k,T} = \sum_{i=k}^{k+T} \gamma^{i-k} r_i, \tag{16}$$

where $0 < \gamma \leq 1$ is the discount factor, which ensures that $J_{k,T}$ remains bounded.

We then formulate a value function as in Equation (17). The value of a policy is defined as the conditional expected value of future cost when in state x at starting time k and following policy $\pi(x,u)$ thereafter [7,8]:

$$V_k^{\pi}(x) = E_{\pi}\{J_{k,T}|x_k = x\} = E_{\pi}\left\{\sum_{i=k}^{k+T} \gamma^{i-k} r_i | x_k = x\right\}. \tag{17}$$

Finally, we find the control policy with the lowest cost/value. In other words, we wish to find the optimal policy,

$$\pi^*(x,u) = \arg\min_{\pi} V_k^{\pi}(x) = \arg\min_{\pi} E_{\pi}\left\{\sum_{i=k}^{k+T} \gamma^{i-k} r_i | x_k = x\right\}, \tag{18}$$

which minimizes the value function and the corresponding optimal value, which is the minimum of the value function,

$$V_k^*(x) = \min_{\pi} V_k^{\pi}(x) = \min_{\pi} E_{\pi}\left\{\sum_{i=k}^{k+T} \gamma^{i-k} r_i | x_k = x\right\}. \tag{19}$$

It is shown in [7,8] that the value function $V_k^{\pi}(x)$ can be reformulated as

$$\begin{aligned} V_k^{\pi}(x) &= E_{\pi}\left\{\sum_{i=k}^{k+T} \gamma^{i-k} r_i | x_k = x\right\} \\ &= E_{\pi}\left\{r_k + \gamma \sum_{i=k+1}^{k+T} \gamma^{i-(k+1)} r_i | x_k = x\right\} \\ &= \sum_u \pi(x,u) \sum_{x'} P_{xx'}^u \left[R_{xx'}^u + \gamma E_{\pi}\left\{\sum_{i=k+1}^{k+T} \gamma^{i-(k+1)} r_i | x_{k+1} = x'\right\}\right] \\ &= \sum_u \pi(x,u) \sum_{x'} P_{xx'}^u \left[R_{xx'}^u + \gamma V_{k+1}^{\pi}(x')\right]. \end{aligned} \tag{20}$$

Now, setting the time horizon T to infinity, we obtain the infinite-horizon cost,

$$J_k = \sum_{i=k}^{\infty} \gamma^{i-k} r_i, \tag{21}$$

and the Bellman equation,

$$V^{\pi}(x) = \sum_u \pi(x,u) \sum_{x'} P_{xx'}^u \left[R_{xx'}^u + \gamma V^{\pi}(x')\right]. \tag{22}$$

The Bellman equation consists of the one-step cost, $\sum_u \pi(x,u)\sum_{x'} P^u_{xx'} R^u_{xx'}$, and a current estimate of discounted future costs, $\gamma V^\pi(x')$. Now, assuming ergodicity of the system, the optimal policy, $\pi^*(x,u)$, is deterministic. Hence, we obtain the optimal value or Bellman optimality equation,

$$V^*(x) = \min_u \sum_{x'} P^u_{xx'} \left[R^u_{xx'} + \gamma V^*(x')\right], \tag{23}$$

and the optimal control policy,

$$u^* = \arg\min_u \sum_{x'} P^u_{xx'} \left[R^u_{xx'} + \gamma V^*(x')\right]. \tag{24}$$

2.3.2. The Discrete-Time Bellman Equation

A discrete-time LTI system such as the one described by Equations (1) and (6) can be considered a deterministic MDP. As a result, the deterministic stage costs are defined in terms of the linear quadratic regulator (LQR) [6], leading to the infinite-horizon cost,

$$J_k = \sum_{i=k}^{\infty} r_i = \sum_{i=k}^{\infty} (x_i^T Q x_i + u_i^T R u_i), \tag{25}$$

where $Q = Q^T \geq 0$ and $R = R^T > 0$ are user-defined weighting matrices. The value function can now be formulated as

$$\begin{aligned} V(x_k) &= \sum_{i=k}^{\infty} r_i = \sum_{i=k}^{\infty} (x_i^T Q x_i + u_i^T R u_i) \\ &= x_k^T Q x_k + u_k^T R u_k + \sum_{i=k+1}^{\infty} (x_i^T Q x_i + u_i^T R u_i) \\ &= x_k^T Q x_k + u_k^T R u_k + V(x_{k+1}), \end{aligned} \tag{26}$$

where the second and third formulations in Equation (26) are forms of the discrete-time Bellman equation consisting of a one-step cost term, $(x_k^T Q x_k + u_k^T R u_k)$, and the current estimate of future costs, $V(x_{k+1})$ [7,8].

We can now derive an explicit form of the discrete-time Bellman equation. To begin, assume that the value function at time k is quadratic in terms of the state, $x(k)$, and can be written as

$$V(x_k) = x_k^T P x_k \tag{27}$$

for some matrix, $P = P^T > 0$. We may now combine Equations (26) and (27) to give

$$V(x_k) = x_k^T P x_k = x_k^T Q x_k + u_k^T R u_k + x_{k+1}^T P x_{k+1}. \tag{28}$$

Substituting x_{k+1} from Equation (1) into Equation (28), we obtain

$$x_k^T P x_k = x_k^T Q x_k + u_k^T R u_k + (A x_k + B u_k)^T P (A x_k + B u_k), \tag{29}$$

and applying a stabilizing static feedback control input, $u_k = -K x_k$,

$$x_k^T P x_k = x_k^T Q x_k + x_k^T K^T R K x_k + x_k^T (A - BK)^T P (A - BK) x_k. \tag{30}$$

Now, equating the two sides of Equation (30), we obtain the Bellman equation in the form of a Lyapunov equation as

$$(A - BK)^T P (A - BK) - P + Q + K^T R K = 0. \tag{31}$$

The discrete-time Hamiltonian function [7,8,15] can be formulated using Equation (29) as

$$H(x_k, u_k) = x_k^T Q x_k + u_k^T R u_k + (A x_k + B u_k)^T P (A x_k + B u_k) - x_k^T P x_k \tag{32}$$

and by using the fact that for vector $\vec{x}$ and matrix A, $\frac{\partial A\vec{x}}{\partial \vec{x}} = A$, $\frac{\partial \vec{x}^T A}{\partial \vec{x}} = A^T$, and that for a symmetrical matrix A, $\frac{\partial \vec{x}^T A\vec{x}}{\partial \vec{x}} = 2\vec{x}^T A$, the optimal feedback control input can be found via $\frac{\partial H(x_k,u_k)}{\partial u_k} = 0$ as

$$\widetilde{u}_k^* = -K_k^* x_k = -(R + B^T PB)^{-1} B^T PAx_k. \tag{33}$$

By substituting for u_k into Equation (29), we obtain the familiar form of the discrete-time algebraic Riccati equation (DARE) as

$$A^T PA - P + Q - A^T PB(B^T PB + R)^{-1} B^T PA = 0. \tag{34}$$

2.3.3. The Discrete-Time Linear Optimal Output Regulation Problem

As previously mentioned, the solution to the LOORP should both stabilize the closed-loop system and asymptotically track a reference output signal. The optimal control policy, $u^* = \mu(x)$, that achieves both of these requirements is obtained by solving two optimization problems [9,10,12].

Problem 1. *We solve the static optimization problem of Equation* (35) *in order to find solutions to the regulator equations stated in Equation* (11)*, thus assuring the asymptotic tracking of the reference signal.*

$$\begin{aligned} \min_{(X,U)} \quad & trace(X^T \widetilde{Q} X + U^T \widetilde{R} U) \\ s.t. \quad & XM = AX + BU + D \\ & 0 = CX - E \end{aligned} \tag{35}$$

where $\widetilde{Q} = \widetilde{Q}^T > 0$ *and* $\widetilde{R} = \widetilde{R}^T > 0$.

Problem 2. *We solve the dynamic optimization problem of Equation* (36) *in order to find the optimal feedback control policy,* $\overline{u}^* = -K_x^* \overline{x}(k)$*, thus assuring stability and satisfactory transient performance of the closed-loop system. Problem 2 minimizes the discrete-time Bellman equation of Equation* (26)*.*

$$\begin{aligned} \min_{\widetilde{u}} \quad & V(k) = \sum_{i=k}^{\infty} (\overline{x}_i^T Q \overline{x}_i + \overline{u}_i^T R \overline{u}_i) \\ s.t. \quad & \overline{x}(k+1) = A\overline{x} + B\overline{u} \\ & e(k) = C\overline{x} + D\overline{u} \end{aligned} \tag{36}$$

where, again, $Q \geq 0$ *and* $R > 0$.

Therefore, the optimal static state feedback controller of Equation (8) is obtained by first solving Problem 2 to find K_x^*, then solving Problem 1 to obtain (X, U), from which K_v^* can be obtained via Equation (12). In order to solve Problems 1 and 2, we utilize the RL algorithms of [9,10,12].

In Algorithm 1 of [12], provided that the system dynamics are completely known, a policy iteration algorithm can be used. The iterative form of the Lyapunov equation (31) used for policy evaluation is given by

$$P^{j+1} = (A - BK^j)^T P^{j+1} (A - BK^j) + Q + (K^j)^T RK^j. \tag{37}$$

The iterative optimal feedback gain used for policy improvement can be obtained using Equation (33) as

$$\begin{aligned} u_k^{j+1} &= \arg\min \quad (x_k^T Q x_k + u_k^T R u_k + x_{k+1}^T P^{j+1} x_{k+1}) \\ &= -K^{j+1} x_k \\ &= -(R + B^T P^{j+1} B)^{-1} B^T P^{j+1} A x_k. \end{aligned} \tag{38}$$

Algorithm 1 of [12], or Hewer's algorithm, involves the solution of the DARE at every iterative step. This algorithm can therefore be summarized as (Algorithm 1):

Algorithm 1 Offline Hewer's PI Algorithm with Known System Dynamics

Initialization: Start with a stabilizing control policy K_x^0. For step index j, iterate Steps 1 and 2 until convergence in Step 3.

Step 1: **Policy Evaluation:** Solve for P^{j+1} using

$$P^{j+1} = (A - BK^j)^T P^{j+1} (A - BK^j) + Q + (K^j)^T R K^j.$$

Step 2: **Policy Improvement:** Update the policy using

$$K^{j+1} = -(R + B^T P^{j+1} B)^{-1} B^T P^{j+1} A x_k.$$

Step 3: **Termination:**

$$||P^{j+1} - P^j||_2 \leq \epsilon$$

for some small positive ϵ. Otherwise, increment index j and return to Step 1.

The main drawback of Algorithm 1 is the requirement of complete knowledge of system dynamics. As previously mentioned, we may utilize Algorithm 3 of [12] to obviate the need for knowledge of system dynamics, instead relying on measured state data to solve for an optimal control solution. Hence, LOORP problems 1 and 2 can both be solved via Algorithm 3 of [12].

Using the first equation in Equation (11), we define a Sylvester map [16] $\Omega : \mathbb{R}^{n_x \times n_v} \to \mathbb{R}^{n_x \times n_v}$ as

$$\Omega(X) = XM - AX. \tag{39}$$

Now, using the second equation in Equation (11) and assuming that the matrix $D = 0$, we can find a suitable form for the solution, X, to this equation by first selecting a constant matrix $X_1 \in \mathbb{R}^{n_x \times n_v}$ such that $CX_1 + F = 0$. Next, we select $X_i \in \mathbb{R}^{n_x \times n_v}$, $i \in 2, 3, \ldots, m+1$, such that the vectors $\text{vec}(X_i)$, $\forall X_i$ form a basis for $\ker(I_{n_v} \otimes C)$, where $m = (n_x - n_r)n_v$ is the dimension of $\ker(I_{n_v} \otimes C)$. Therefore, $CX_i = 0$, $\forall X_i$ and X can be written as

$$X = X_1 + \sum_{i=2}^{m+1} \alpha_i X_i, \quad \alpha_i \in \mathbb{R}. \tag{40}$$

We now utilize the error system dynamics of Equation (15) and matrices X_i from Equation (40) to define a new state $\overline{x}_i(k)$ as

$$\overline{x}_i(k) = x(k) - X_i v(k), \tag{41}$$

and using Equations (6) and (41), we define the dynamics of this new state as

$$\begin{aligned} \overline{x}_i(k+1) &= x(k+1) - X_i v(k+1) \\ &= Ax(k) + Bu(k) + (E - X_i M) v(k). \end{aligned} \tag{42}$$

By reformulating Equation (41) as $x(k) = \overline{x}_i(k) + X_i v(k)$ and substituting into Equation (42), we obtain

$$\overline{x}_i(k+1) = A\overline{x}_i(k) + Bu(k) + (E - X_iM + AX_i)v(k). \tag{43}$$

Then, by adding and subtracting the term $B\overline{u}_i(k) = -BK_x^j\overline{x}_i(k)$ and using Equation (39), we obtain

$$\overline{x}_i(k+1) = (A - BK_x^j)\overline{x}_i(k) + B(u(k) + K_x^j\overline{x}_i(k)) + (E - \Omega(X_i))v(k). \tag{44}$$

We now define $\pi(X_i) = \Omega(X_i) - E$ and substitute into Equation (44) to give

$$\overline{x}_i(k+1) = (A - BK_x^j)\overline{x}_i(k) + B(u(k) + K_x^j\overline{x}_i(k)) - \pi(X_i)v(k). \tag{45}$$

From Equation (28), we obtain

$$\overline{x}_i^T(k)Q\overline{x}_i(k) + \overline{u}_i^T(k)R\overline{u}_i(k) = \overline{x}_i^T(k+1)P^{j+1}\overline{x}_i(k+1) - \overline{x}_i^T(k)P^{j+1}\overline{x}_i(k). \tag{46}$$

By substituting $\overline{x}_i(k+1)$ from Equation (45) into Equation (46), we obtain a system of equations that forms the foundation of RL algorithms that solve for optimal control solutions with unknown system dynamics [9,10,12].

As per [12] and using properties of the Kronecker product, we can now define the following:

$$\begin{aligned} L_1^{j+1} &= A^TP^{j+1}B \\ L_2^{j+1} &= B^TP^{j+1}B \\ L_{3i}^{j+1} &= \pi(X_i)^TP^{j+1}\pi(X_i) \\ L_{4i}^{j+1} &= A^TP^{j+1}\pi(X_i) \\ L_{5i}^{j+1} &= B^TP^{j+1}\pi(X_i) \end{aligned} \tag{47}$$

$$\phi_i^j(k) = \begin{bmatrix} (\overline{x}_i^T(k) \otimes \overline{x}_i^T(k))\mathrm{vec}\left(-Q - (K_x^j)^TRK_x^j\right) \\ (\overline{x}_i^T(k+1) \otimes \overline{x}_i^T(k+1))\mathrm{vec}\left(-Q - (K_x^j)^TRK_x^j\right) \\ \vdots \\ (\overline{x}_i^T(k+s) \otimes \overline{x}_i^T(k+s))\mathrm{vec}\left(-Q - (K_x^j)^TRK_x^j\right) \end{bmatrix} \tag{48}$$

where $s \geq [\,([n_x \times (n_x+1)]/2) + ([n_u \times (n_u+1)]/2) + ([n_v \times (n_v+1)]/2) + n_x(n_u + n_v) + (n_u \times n_v) - 1]$,

$$\psi_i^j(k) = \begin{bmatrix} \Phi_{11} & \Phi_{12} & \Phi_{13} & \cdots & \Phi_{16} \\ \Phi_{21} & \Phi_{22} & \Phi_{23} & \cdots & \Phi_{26} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \Phi_{s1} & \Phi_{s2} & \Phi_{s3} & \cdots & \Phi_{s6} \end{bmatrix} \tag{49}$$

where

$$\begin{aligned}
\Phi_{l1} &= \left(\overline{x}_i^T(k+l+1) \otimes \overline{x}_i^T(k+l+1)\right) - \left(\overline{x}_i^T(k+l) \otimes \overline{x}_i^T(k+l)\right) \\
\Phi_{l2} &= -2\left(\left(K_x^j \overline{x}_i(k+l) + u(k+l)\right)^T \otimes \overline{x}_i^T(k+l)\right) \\
\Phi_{l3} &= -\left(\left(K_x^j \overline{x}_i(k+l) + u(k+l)\right)^T \otimes \left(-K_x^j \overline{x}_i(k+l) + u(k+l)\right)^T\right) \\
\Phi_{l4} &= -v^T(k+l) \otimes v^T(k+l) \\
\Phi_{l5} &= 2\left(v^T(k+l) \otimes \overline{x}_i^T(k+l)\right) \\
\Phi_{l6} &= 2\left(v^T(k+l) \otimes u^T(k+l)\right)
\end{aligned} \tag{50}$$

Equations (47)– (50) can be used as in [12] to obtain the linear equation

$$\psi_i^j(k)\left[P^{j+1}, L_1^{j+1}, L_2^{j+1}, L_{3i}^{j+1}, L_{4i}^{j+1}, L_{5i}^{j+1}\right] = \phi_i^j(k), \tag{51}$$

which can be solved using least squares to obtain

$$\left[P^{j+1}, L_1^{j+1}, L_2^{j+1}, L_{3i}^{j+1}, L_{4i}^{j+1}, L_{5i}^{j+1}\right] = ([\psi_i^j(k)]^T \psi_i^j(k))^{-1} [\psi_i^j(k)]^T \phi_i^j(k). \tag{52}$$

The feedback gain is updated during the policy improvement step of the RL algorithm as

$$K_x^{j+1} = (R + L_2^{j+1})^{-1} (L_1^{j+1})^T. \tag{53}$$

which iteratively converges to the optimal feedback gain, K_x^*, hence solving Problem 2. The least-squares solution from Equation (52) is used to obtain a solution to Problem 1 as follows.

Firstly, let

$$\begin{aligned}
\overline{\Lambda} &= \begin{bmatrix} \mathrm{vec}(L_{42}^{j+1} - L_{40}^{j+1}) & \cdots & \mathrm{vec}(L_{4(m+1)}^{j+1} - L_{40}^{j+1}) & 0 & -I_{n_v} \otimes L_1^{j+1} \\ \mathrm{vec}(X_2) & \cdots & \mathrm{vec}(X_{m+1}) & -I_{n_x \times n_v} & 0 \end{bmatrix} \\
&= \begin{bmatrix} \overline{\Lambda}_{11} & \overline{\Lambda}_{12} \\ \overline{\Lambda}_{21} & \overline{\Lambda}_{22} \end{bmatrix}
\end{aligned} \tag{54}$$

and

$$\overline{\xi} = \begin{bmatrix} \overline{\xi}_1 \\ \overline{\xi}_2 \end{bmatrix} = \begin{bmatrix} \mathrm{vec}(-L_{41}^{j+1}) \\ -\mathrm{vec}(X_1) \end{bmatrix}. \tag{55}$$

Now, define

$$\overline{\Pi} = -\overline{\Lambda}_{11} \overline{\Lambda}_{21}^{-1} \overline{\Lambda}_{22} + \overline{\Lambda}_{12} \tag{56}$$

and

$$\overline{\Psi} = -\overline{\Lambda}_{11} \overline{\Lambda}_{21}^{-1} \overline{\xi}_2 + \overline{\xi}_1. \tag{57}$$

We can now obtain matrices (X, U) via least squares by solving the equation

$$\overline{\Pi} \begin{bmatrix} \mathrm{vec}(X) \\ \mathrm{vec}(U) \end{bmatrix} = \overline{\Psi}. \tag{58}$$

The off-policy RL algorithm for solving both Problems 1 and 2 can be summarized as (Algorithm 2)

Algorithm 2 Data-driven Algorithm for Iterative Solution of LOORP

Initialization: Bring the system to steady state by utilizing a stabilizing initial control policy, $\hat{u} = u + \hat{e}(k) = K_x^j x(k) + \hat{e}(k)$, where $\hat{e}(k)$ is injected perturbation noise, $j = 0$, and $i = 0$. Also, supply weighting matrices Q and R.

Problem 1:

Step 1: **Policy Evaluation:** Use circular buffers to collect samples of state, input, and reference signals in order to build matrices $\phi_i^j(k)$ and $\psi_i^j(k)$. Solve for $\left[P^{j+1}, L_1^{j+1}, L_2^{j+1}, L_{3i}^{j+1}, L_{4i}^{j+1}, L_{5i}^{j+1}\right]$ using Equation (52).

Step 2: **Policy Improvement:** Update the control policy using Equation (53):

$$K_x^{j+1} = (R + L_2^{j+1})^{-1}(L_1^{j+1})^T.$$

Step 3: **Termination:** If $||K_x^{j+1} - K_x^j||_2 \leq \epsilon$ for some small positive constant ϵ, go to **Step** 4. Otherwise, set $j = j + 1$ and return to **Step** 1.

Problem 2:

Step 4: Set $j = j^*$ and $i = i + 1$ and solve for L_{4i}^{j+1} using Equation (52), repeating this step until $i = m + 1$.

Step 5: Find matrices (X, U) by solving Equation (58).

Solution: Using the solutions to Problems 1 and 2, we obtain the optimal control solution, $u^*(k)$, from Equations (8) and (12) as

$$K_v^* = U + K_x^* X$$
$$u^*(k) = -K_x^* x(k) + K_v^* v(k).$$

Stop.

2.4. System Architecture

In order to assess the performance of the proposed model-free RL algorithm, it was incorporated into the simulation of a closed-loop power electronic system consisting of a buck converter as the system plant. An ideal buck converter, without parasitics, was utilized. Considering parasitics, such as the equivalent series resistance (ESR) of the capacitor, r_C, would mainly add a high-frequency left-half-plane zero to the converter dynamics. Being at high frequency, $\frac{1}{Cr_C}$, this zero would not significantly alter the resulting controller design ([17], p. 410).

The schematic of the buck converter is depicted in Figure 4. The buck converter is a pulse-width-modulated(PWM), DC-DC, non-isolated power electronic converter topology that steps-down/bucks an input voltage down to a lower output voltage [17]. The circuit parameters for the simulated buck converter plant are given in Table 1.

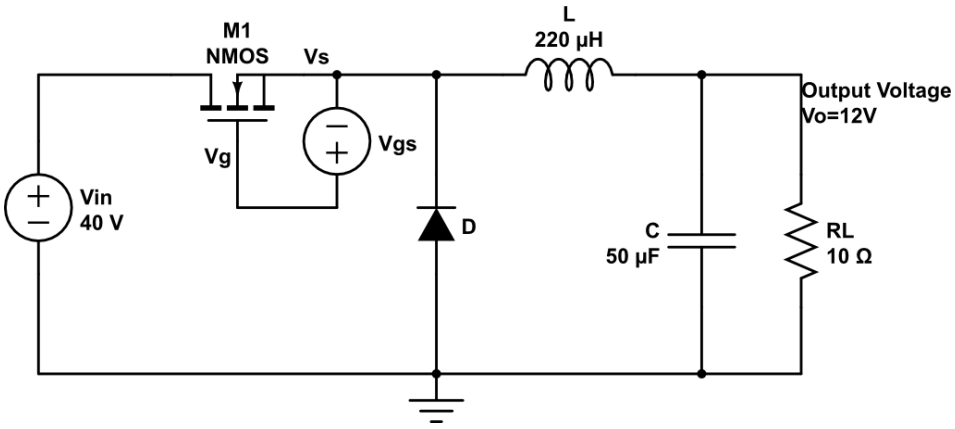

Figure 4. Schematic of buck converter topology.

Table 1. Circuit parameters for buck converter plant.

Parameter	Value
Filter Capacitance, C/ESR	50 µF/0 Ω
Filter Inductance, L/ESR	220 µH/0.1 Ω
Switching Frequency, f_s	50–500 kHz
Input Voltage, v_{in}	40 V
Output Voltage, v_o	12 V
Output Resistance, R_L	10 Ω

Simulations were carried out using Simulink. All simulations were run at an arbitrary switching frequency of 80 kHz before being re-run at switching frequencies ranging from 50 to 500 kHz, hence assuring the resilience of the RL algorithm to switching frequency changes. The Simulink model of the buck power converter simulation is shown in Figure 5. Algorithm 2 is implemented in the DSP subsystem of the model. An in-depth view of the algorithm implementation is shown in Figure 6.

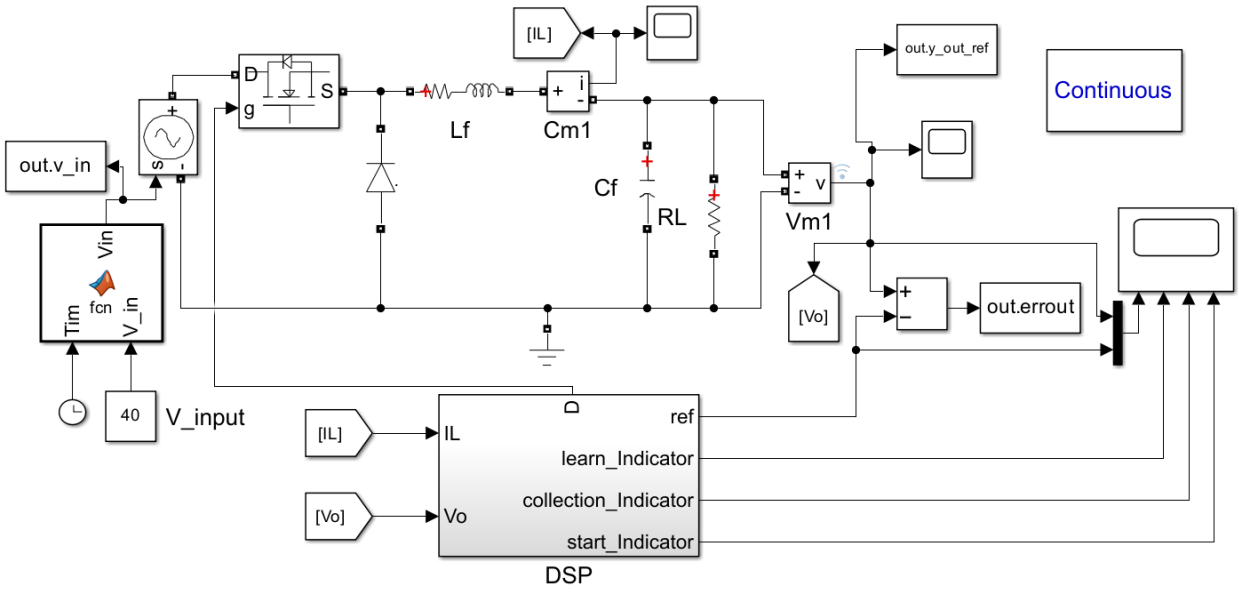

Figure 5. Simulink model used for buck converter simulations.

As seen in Figure 6, Algorithm 2 is implemented using three subsystems, namely, the *Collection*, *Learning* and *Learnt* subsystems, with an additional *Relearn* subsystem at the top of the figure. Algorithm 2 is able to learn an optimal control solution at a specific operating point. In order to allow the system to respond to changes in operating conditions (line/load variations), a relearning mechanism was developed to learn a new optimal control solution if an error threshold was surpassed [13]. This means that the relearning mechanism deployed in the *Relearn* subsystem allows the system to respond to errors in the output just as a classical PID controller would. Additionally, provided that the relearning algorithm was able to converge to new control solutions fast enough, the performance of the RL-based controller could potentially be on par with that of a classical PID controller.

Data memory stores are used to transfer data, such as intermediate matrices and gains, among the various subsystems with each subsystem containing code-based, discrete-time MATLAB function blocks. The switch in the upper-right quadrant of Figure 6 is used to switch between the initial input $\hat{u}$ of Algorithm 2 and the learnt optimal control input $u^*(k)$. As detailed in the Initialization of Algorithm 2, $\hat{u}$ is formed from the combination of the initial input u and injected random noise $\hat{e}(k)$, with initial input u being supplied via the data memory store read *U_pert*. Also, the user-defined weighting matrices Q and R are given as inputs *Qbar* and *Rbar*, respectively, to the *Learning* subsystem.

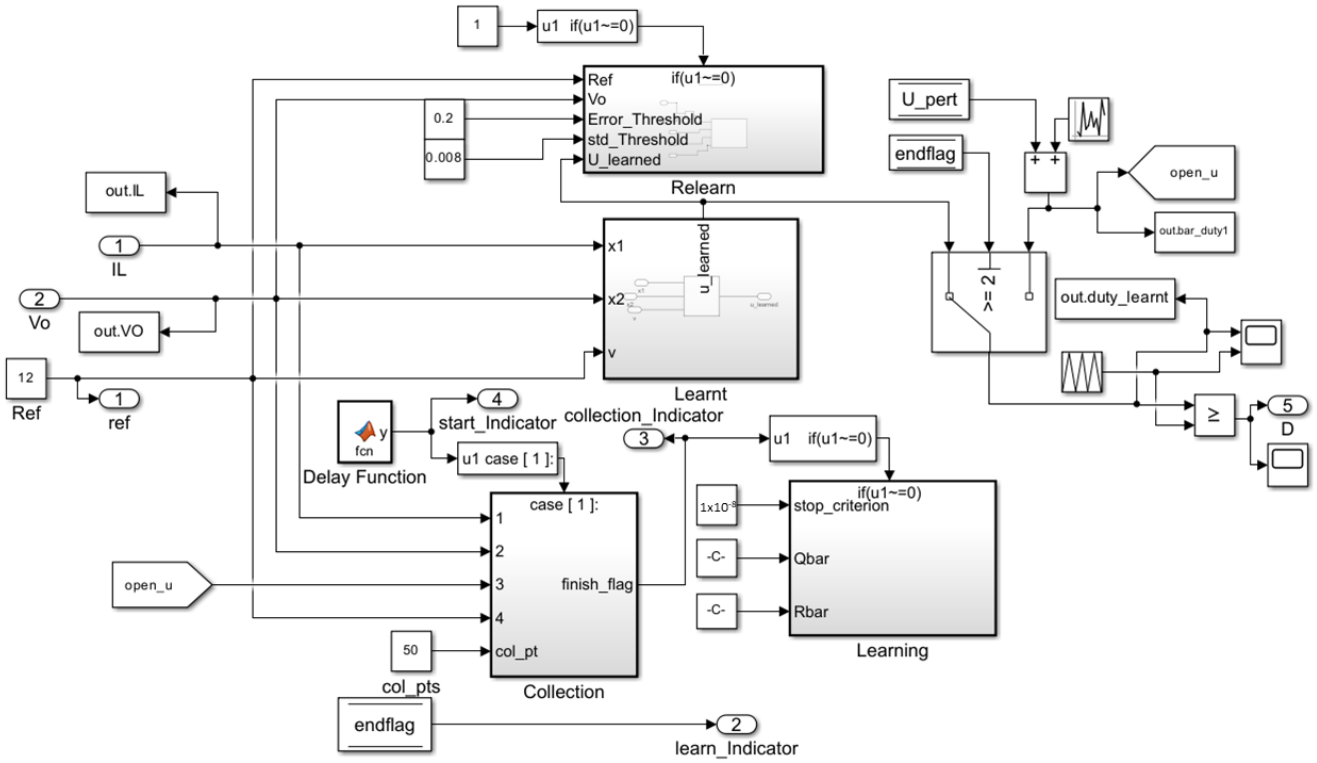

Figure 6. Implementation of RL algorithm in Simulink model.

The *Delay Function* block is used, as the name suggests, to create a delay that allows for the system to reach steady-state before the collection of data commences. The *Collection* subsystem is then used to implement circular buffers to collect samples of state, input, and reference signals in order to build matrices $\phi_i^j(k)$ and $\psi_i^j(k)$, as described in Step 1 of Algorithm 2. The lengths of these buffers is given by the *col_pts* constant block. Once the required matrices have been formulated, we solve for $\left[P^{j+1}, L_1^{j+1}, L_2^{j+1}, L_{3i}^{j+1}, L_{4i}^{j+1}, L_{5i}^{j+1}\right]$ using the *Learning* subsystem. Steps 2–5 of Algorithm 2 are also implemented in the *Learning* subsystem, from which we obtain the optimal gains K_x^* and K_v^*. Finally, we obtain the optimal control input by implementing the equation $u^*(k) = -K_x^* x(k) + K_v^* v(k)$ in the *Learnt* subsystem and switch to using $u^*(k)$ as the input to the buck converter plant.

2.5. Design of Classical Controller

A conventional Type-III compensator was designed using standard methods to serve as a comparison for the performance of the RL algorithm presented. A detailed design procedure for such a compensator can be found in [18]. The Type-III compensator is designed to provide a boost in phase at a desired crossover frequency, as depicted in the Bode plot in Figure 7. This differs from the tuning methods of a PID controller, where gains are adjusted to meet transient performance requirements. The transfer function of the small-signal buck converter model was derived using the state-space averaging method as

$$G(s) = \frac{4.364 \times 10^9}{s^2 + 2459s + 9.183 \times 10^7}. \tag{59}$$

The corresponding Bode plot of the buck converter plant is depicted in Figure 8.

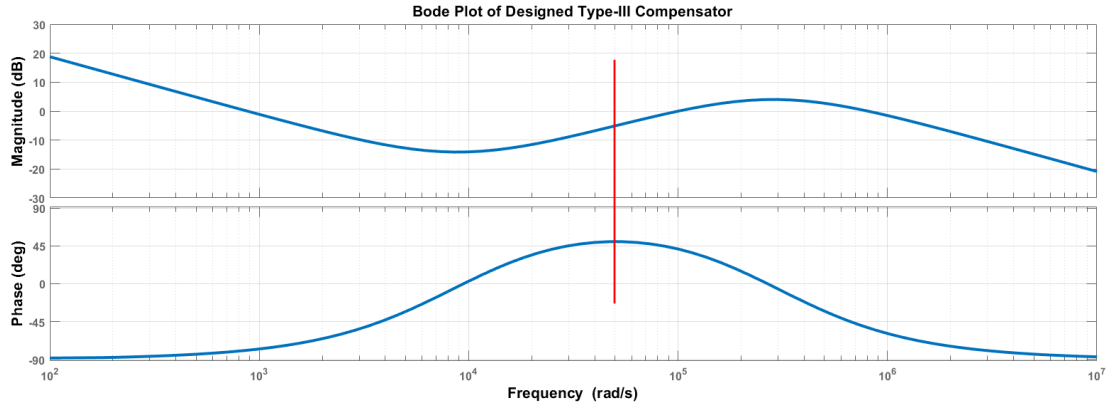

Figure 7. Bode plot of Type-III compensator.

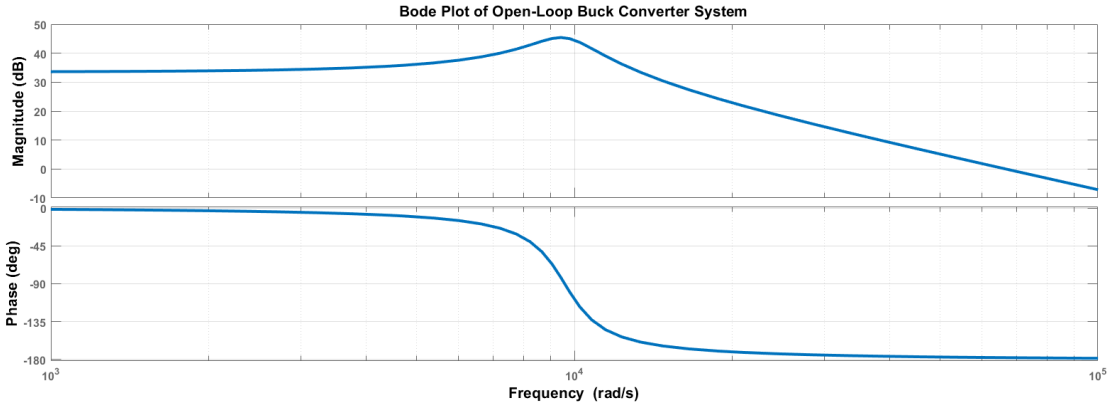

Figure 8. Bodeplot of buck converter plant/system.

The Type-III compensator was designed with a desired crossover frequency of 8 kHz or $\sim$ 50 krad/s and a phase margin of 53°. The transfer function of the designed compensator is given in Equation (60) and the corresponding Bode plot in Figure 7.

$$C(s) = \frac{9.076 \times 10^5 s^2 + 1.605 \times 10^{10} s + 7.095 \times 10^{13}}{s^3 + 5.715 \times 10^5 s^2 + 8.165 \times 10^{10} s} \tag{60}$$

The Bode plot of the controller–plant combination is shown in Figure 9. The vertical red line indicates the crossover frequency is at roughly 50 krad/s as designed.

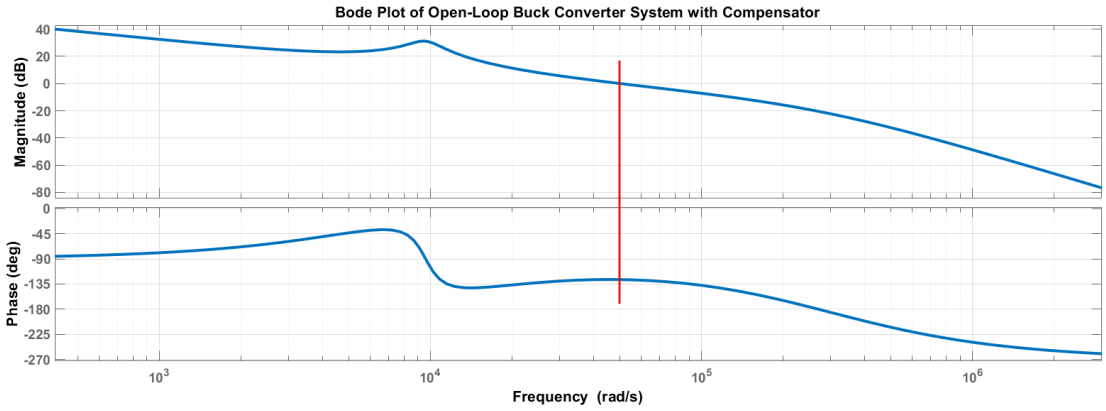

Figure 9. Bode plot of controller–plant combination.

3. Results

Figure 10 depicts the line regulation ability of the RL algorithm with relearning enabled. The output voltage waveform is shown in blue and the one-third-scale input voltage waveform in red. As described in [13], the start-up transient occurs in Period 1 and data samples are collected and the optimal control solution corresponding to the 40 V input voltage are learnt during Period 2 before being applied in Period 3. At the beginning of Period 4, the input voltage increases to 48 V and the relearning process is triggered. Hence, the control input is changed to $\hat{u}$ and data samples are collected before learning a new control solution. The new control solution corresponding to an input voltage of 48 V is applied in Period 5 and the output voltage returns to the reference value of 12 V. The system response to additional step-changes in the input voltage to 60 V and 36 V is also depicted in Periods 6–9.

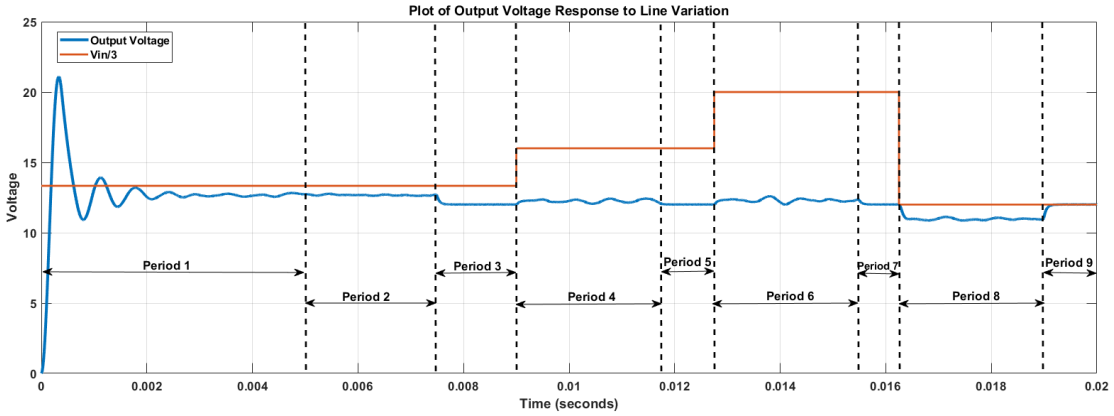

Figure 10. Plot of output voltage response to line variation.

The system's ability to respond to load variations is depicted in Figure 11. The system is allowed to reach a steady state before data collection and learning begins at $t = 0.00425$ s. The learnt solution is then implemented at $t = 0.00625$ s. The load then changes to 5 Ω at $t = 0.007$ s and the relearning mechanism triggered. The input is changed to $\hat{u}$, data samples are collected, and the optimal control solution is learnt. The new learnt optimal solution is then implemented at $t = 0.0085$ s. At $t = 0.0125$ s, the load changes to 10 Ω and the relearning mechanism is triggered. Again, $\hat{u}$ is used and data samples are collected before the new optimal control solution is learnt. Finally, the learnt solution is implemented at $t = 0.014$ s. Figures 10 and 11 show that the RL algorithm is able to successfully respond to line and load variations. Next, we compare the performance of the RL algorithm to that of a classical controller.

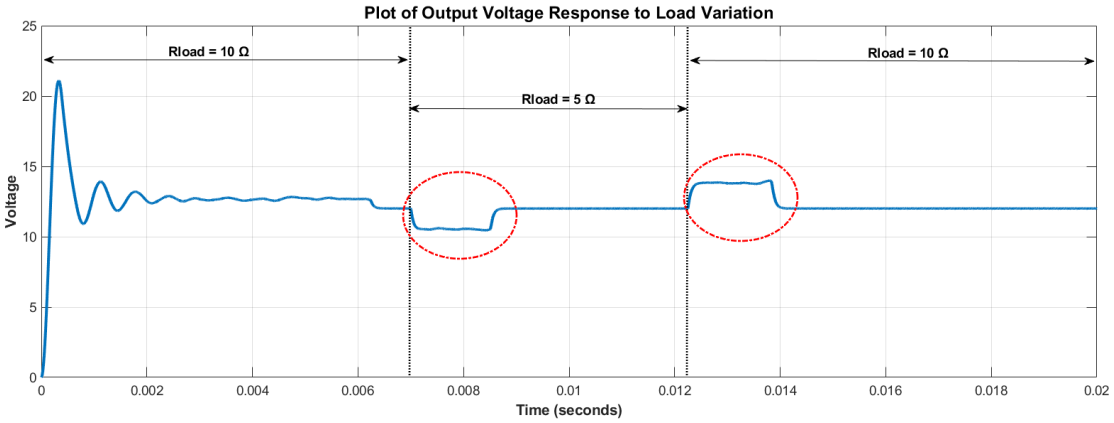

Figure 11. Plot of output voltage response to load variation.

Figure 12 compares the line regulation ability of the proposed RL algorithm to that of the designed Type-III compensator. The one-third-scale input voltage waveform is depicted in blue, the output voltage response of the system using the RL algorithm is depicted in red, and the output voltage response of the system with the Type-III compensator is depicted in yellow. From the figure, it is clear that a conventional controller has a faster response time than the RL algorithm with relearning enabled. Hence, in our previous work [13], we concluded that the RL algorithm is feasible but does not surpass the performance of conventional compensators if deployed using the proposed relearning mechanism. The RL algorithm's response time is impacted by the number of data samples collected, the sampling period, and the number of iterations until convergence. On the other hand, the classical controller was designed to have a crossover frequency of 8 kHz and relatively fast response times.

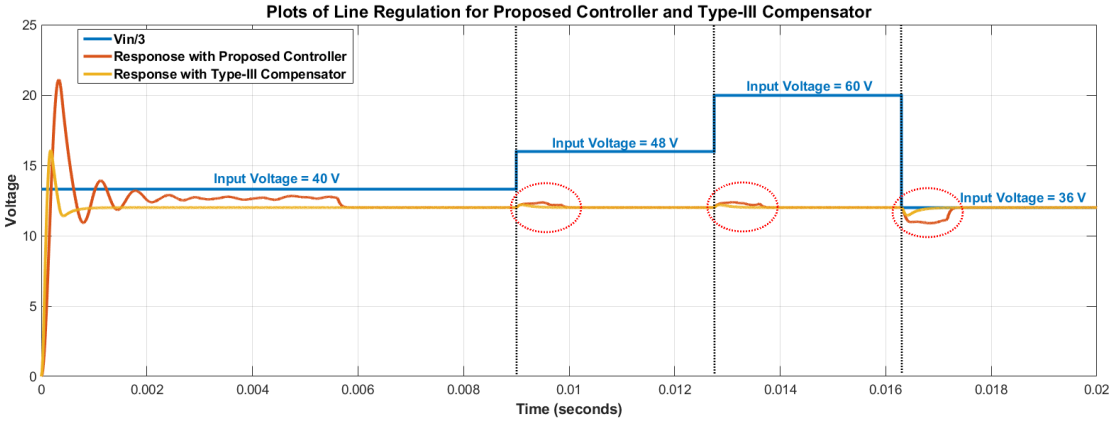

Figure 12. Comparison of line regulation performance of RL algorithm and Type-III compensator.

After conducting further research, a different approach to implementing the RL algorithm was developed. Firstly, it was observed that the response time of the RL algorithm was being severely impacted by the collection of data samples and the iterative learning process. By storing the optimal control gains learnt at a previous point in time for a particular operating point and applying these gains as soon as the system starts up, the output voltage is able to quickly settle at the reference voltage. Figure 13 provides clear evidence of the feasibility of this approach. Depicted are the output voltage step responses of the RL algorithm in red and the designed Type-III compensator in blue with a load of 10 Ω and input voltage of 40 V. The figure shows that the step-response transient characteristics of the RL-based controller using pre-learnt gains surpasses that of the conventional controller. In particular, the pre-learnt controller has less overshoot and a faster settling time. Specifically, the overshoot decreased from 35% to 23% and the settling time decreased by 0.3 s.

In order to expand the pre-learnt approach over several operating points, learnt gains for a vast range of operating points can either be stored in a look-up table (LUT) or functionally mapped over a control surface. This will, however, require the input voltage and load conditions to be constantly monitored in order to map the current operating point to the corresponding control gains. Storing gains in a large LUT and using interpolation and extrapolation techniques would produce an accurate control solution; however, this would not be easily implemented on a run-of-the-mill microcontroller. Alternatively, by fitting a nonlinear polynomial function to the pre-learnt gains, the required gains for a particular operating point can be easily calculated and applied. This approach is easy to implement on a microcontroller; however, it introduces fitting errors, which manifest as small errors in the output voltage level. Provided that these errors are small and acceptable to the user, nonlinear gain scheduling can be utilized to make the RL algorithm more competitive.

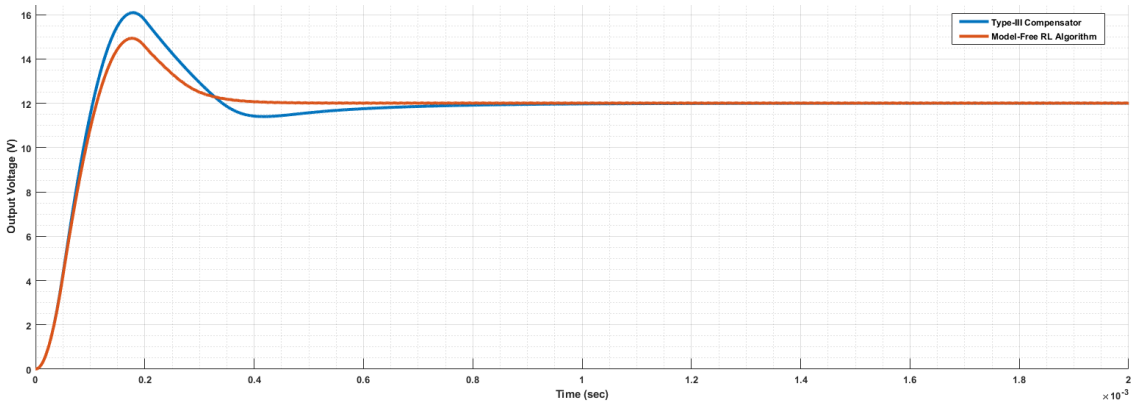

Figure 13. Comparison of step response of pre-learnt gains from RL algorithm and Type-III compensator.

Having taken the exosystem (reference) dynamics into account while formulating the model-free RL algorithm, it is expected that the closed-loop system should asymptotically track a given reference signal. This tracking ability is depicted in Figure 14. Using the same optimal controller gains

$$K_x^* = \begin{bmatrix} 0.2545 \\ 0.1327 \end{bmatrix}$$
$$K_v^* = 0.1834$$

at all reference levels, the reference voltage is varied from 6 to 30 V. The system is able to successfully track the reference with no tracking error, provided that the line and load conditions remain constant.

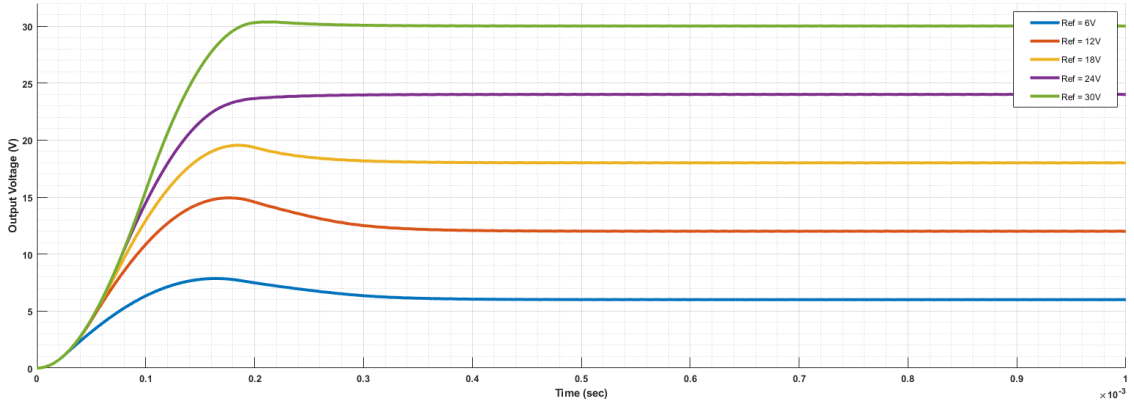

Figure 14. Reference tracking performance of RL algorithm.

4. Discussion

This paper focuses on a novel RL-based control algorithm for power electronic converters. A key benefit of the proposed algorithm is that the controller does not require knowledge of the system dynamics. Additionally, although the system under investigation is a single-input, single-output (SISO) system, the algorithm is just as applicable to linear discrete-time multiple-input, multiple-output (MIMO) systems. The detailed derivation of the RL control technique is provided in Section 2, with the RL algorithm summarized as Algorithm 2.

Figure 6 presents the implementation of the RL algorithm in Simulink. It was shown that this RL-based control algorithm was able to achieve line and load regulation; however, the performance of the proposed controller was shown to be inferior to a more conventional power electronic controller when using the relearning mechanism. This relearning mechanism was developed as a means of allowing the RL algorithm to mimic the behavior of conventional closed-loop controllers; however, due to having to change

control inputs, collect data samples, and allow the algorithm to converge, the response time of the RL algorithm with relearning enabled is much slower than that of conventional PID-based controllers.

The RL algorithm was tested at several operating points and it was observed that the algorithm would converge to a control input of $u^*(k) = 0$ when the inductor current, i_L, entered discontinuous conduction mode (DCM). This makes theoretical sense, as the underlying value function of the RL algorithm is an energy minimization function acting on the states and inputs of the system. The minimum energy state of the inductor current in DCM is 0; hence, the algorithm converges to the zero state. Extension of the algorithm into the DCM region is currently being investigated. While the inductor current is in continuous conduction mode, CCM, the RL algorithm is able to successfully converge to the optimal control solution. However, the relationship between the control gains and the inductor current is nonlinear and the gain function "levels-off"/flattens as the inductor current increases. Therefore, there is a maximum achievable control gain that limits the operating range of the algorithm. This control gain limit appears to be dependent on the perturbation noise frequency and amplitude as well as on the values of the Q and R weighting matrices. This means that the operating range is dependent on one end on DCM and on the other end on the variable maximum achievable control gain.

The optimal controller gains vary based on the operating point of the converter. The operating point for this converter refers to the input voltage and load conditions. By pre-learning controller gains over a wide range of operating points and fitting a multi-variable polynomial function to the operating surface, the RL algorithm is able to respond much faster to changes in operating conditions. It is shown in Figure 13 that the step-response of the RL algorithm using pre-learnt gains surpasses the conventional controller in terms of transient performance. Inaccuracies in the fitting function would, however, lead to small stead-state errors.

By deploying this RL algorithm on a physical control system, the user would only have to supply weighting matrices Q and R and start with a stabilizing control input $\hat{u}$ in order to achieve closed-loop control of their power electronic converter. This is a significant improvement to more involved, conventional methods, which require measurement and analysis of the frequency-domain characteristics of the physical system or its linear model in order to design a closed-loop controller. This would mean that a power electronic engineer would only be concerned with designing the hardware to meet open-loop requirements and all closed-loop control would be handled automatically. The learning process of the RL algorithm, however, is computationally intensive, which impacts the ease of deployment of the algorithm. Deployment of the RL algorithm is currently being investigated and experimental validation of the simulated controller will be presented in a future paper.

Future work will involve experimental verification of the findings of this paper. This work will also be extended to other power electronic converters whose accurate models are difficult to obtain. These converters include grid-connected inverters and resonant converters in wireless power transfer (WPT) systems.

5. Conclusions

This paper presents an in-depth analysis of the mathematical theory of reinforcement learning-based control systems. While our previous work [13] presented RL-based control as a novel control strategy for power electronics, this paper expounds on the underlying mathematical theory used to develop reinforcement learning control techniques.

Additionally, an alternative to the relearning mechanism introduced in our previous paper is presented. While the relearning mechanism was not able to compete with the performance of a conventional PID-based controller, the pre-learnt-gains approach presented in this paper provides an avenue through which the developed RL-based control algorithm can potentially outperform more conventional controllers. This was achieved by pre-learning gains over a wide range of operation points and fitting a multi-variable polynomial function to the operating surface. This allows the required gains to be calculated

based on the current operating point of the system and applied to obtain the optimal control input. This approach is not very computationally intensive and can easily be implemented on a suitable microcontroller.

The limitation of using this nonlinear gain-scheduling technique is that small fitting errors introduced by the nonlinear polynomial function lead to small steady-state errors in the output. An additional limitation of the RL-based control presented is the computationally intensive nature of the matrix manipulations in the RL algorithm. This limits the learning algorithm to be only suitable for deployment on advanced microprocessors. The pre-learning of gains aims to address this but with the trade-off of small errors in the steady state.

Finally, the sensitivity of the algorithm to the amplitude and frequency of injected perturbation noise and the values of the Q and R weighting matrices limit the ability to generalize RL-based control. This issue is currently under investigation. These parameters are currently tuned via trial and error to find values that allow for control over the desired operating surface. Improved methods based on system analytics are currently being pursued. One improvement being considered is the modification of the underlying value function used to derive the RL algorithm such that steady-state error and heat loss are also minimized.

Author Contributions: Conceptualization, D.C. and D.A.; methodology, D.A.; software, D.A and J.T.; validation, D.A., D.C., and J.T.; formal analysis, D.A.; investigation, D.A.; resources, D.A. and J.T.; data curation, D.A.; writing—original draft preparation, D.A.; writing—review and editing, D.A.; visualization, D.A.; supervision, D.C.; project administration, D.C.; funding acquisition, D.C. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: Research files available at https://github.com/dajralfred/RL_Control (accessed on 2 February 2024).

Acknowledgments: The authors would like to thank the NYU Tandon ECE department for their unwavering support of our research.

Conflicts of Interest: Jiaxin Teng was employed by Texas Instruments. The remaining authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest. Texas Instruments had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript, or in the decision to publish the results.

Abbreviations

The following abbreviations are used in this manuscript:

RL	Reinforcement Learning
PID	Proportional Integral Differential
LQR	Linear Quadratic Regulator
ARE	Algebraic Riccati Equation
UNCCC	United Nations Climate Change Conference
COP	Conference of the Parties
DER	Distributed Energy Resource
EV	Electric Vehicle
DC	Direct Current
AC	Alternating Current
HF	High-Frequency
HJB	Hamilton–Jacobi–Bellman
RLS	Recursive Least Squares
LCL	Inductor–Capacitor–Inductor
DRL	Deep Reinforcement Learning
NPC	Neutral-Point Clamped
DARE	Discrete-time Algebraic Riccati Equation

DT	Discrete Time
LTI	Linear Time-Invariant
LOORP	Linear Optimal Output Regulation Problem
MDP	Markov Decision Process
ESR	Equivalent Series Resistance
PWM	Pulse-Width Modulation
SISO	Single-Input, Single-Output
MIMO	Multiple-Input, Multiple-Output
DCM	Discontinuous Conduction Mode
CCM	Continuous Conduction Mode
WPT	Wireless Power Transfer
MDPI	Multidisciplinary Digital Publishing Institute

References

1. Alayza, N.; Bhandari, P.; Burns, D.; Cogswell, N.; de Zoysa, K.; Finch, M.; Fransen, T.; González, M.L.; Krishnan, N.; Langer, P.; et al. COP27: Key Takeaways and What's Next. 2022. Available online: https://www.wri.org/insights/cop27-key-outcomes-un-climate-talks-sharm-el-sheikh?utm_source=twitter&utm_medium=anidasguptawri&utm_campaign=socialmedia&utm_term=bac9ea7d-4218-43d3-9dba-bc822c6d1291 (accessed on 2 February 2024).
2. Basu, A.K.; Chowdhury, S.; Chowdhury, S.; Paul, S. Microgrids: Energy management by strategic deployment of DERs—A comprehensive survey. *Renew. Sustain. Energy Rev.* **2011**, *15*, 4348–4356. [CrossRef] [CrossRef]
3. Aguero, J.R.; Takayesu, E.; Novosel, D.; Masiello, R. Modernizing the Grid: Challenges and Opportunities for a Sustainable Future. *IEEE Power Energy Mag.* **2017**, *15*, 74–83. [CrossRef] [CrossRef]
4. Khatibi, M.; Ahmed, S. Impact of distributed energy resources on frequency regulation of the bulk power system. In Proceedings of the 2019 IEEE Conference on Power Electronics and Renewable Energy (CPERE), Aswan, Egypt, 23–25 October 2019; pp. 258–263.
5. Kazimierczuk, M.K.; Czarkowski, D. *Resonant Power Converters*; John Wiley & Sons: Hoboken, NJ, USA, 2012.
6. Lewis, F.L.; Vrabie, D.; Syrmos, V.L. *Optimal Control*; John Wiley & Sons: Hoboken, NJ, USA, 2012.
7. Vrabie, D.; Vrabie, D.; Vamvoudakis, K.; Lewis, F. *Optimal Adaptive Control and Differential Games by Reinforcement Learning Principles*; Control, Robotics and Sensors; Institution of Engineering and Technology: Herts, UK, 2013.
8. Lewis, F.L.; Vrabie, D.; Vamvoudakis, K.G. Reinforcement learning and feedback control: Using natural decision methods to design optimal adaptive controllers. *IEEE Control. Syst. Mag.* **2012**, *32*, 76–105.
9. Gao, W.; Jiang, Z.P. Adaptive dynamic programming and adaptive optimal output regulation of linear systems. *IEEE Trans. Autom. Control.* **2016**, *61*, 4164–4169. [CrossRef] [CrossRef]
10. Gao, W.; Jiang, Z.P. Linear optimal tracking control: An adaptive dynamic programming approach. In Proceedings of the 2015 American Control Conference (ACC), Chicago, IL, USA, 1–3 July 2015; pp. 4929–4934. . [CrossRef] [CrossRef]
11. Qashqai, P.; Babaie, M.; Zgheib, R.; Al-Haddad, K. A Model-Free Switching and Control Method for Three-Level Neutral Point Clamped Converter Using Deep Reinforcement Learning. *IEEE Access* **2023**, *11*, 105394–105409. [CrossRef] [CrossRef]
12. Jiang, Y.; Kiumarsi, B.; Fan, J.; Chai, T.; Li, J.; Lewis, F.L. Optimal output regulation of linear discrete-time systems with unknown dynamics using reinforcement learning. *IEEE Trans. Cybern.* **2019**, *50*, 3147–3156. [CrossRef] [PubMed] [CrossRef] [PubMed]
13. Alfred, D.; Czarkowski, D.; Teng, J. Model-Free Reinforcement-Learning-Based Control Methodology for Power Electronic Converters. In Proceedings of the 2021 IEEE Green Technologies Conference (GreenTech), Denver, CO, USA, 7–9 April 2021; pp. 81–88. [CrossRef]
14. Huang, J. *Nonlinear Output Regulation: Theory and Applications*; SIAM: Philadelphia, PA, USA, 2004.
15. Boyd, S.P.; Vandenberghe, L. *Convex Optimization*; Cambridge University Press: Cambridge, UK, 2004.
16. Wu, A.G.; Duan, G.R.; Xue, Y. Kronecker Maps and Sylvester-Polynomial Matrix Equations. *IEEE Trans. Autom. Control.* **2007**, *52*, 905–910. [CrossRef] [CrossRef]
17. Kazimierczuk, M.K. *Pulse-Width Modulated DC-DC Power Converters*; John Wiley & Sons: Hoboken, NJ, USA, 2015; pp. 470–521.
18. Cao, L. Type III compensator design for power converters. *Power Electron.* **2011**, 20–25. Available online: https://caxapa.ru/thumbs/597674/Type3CompensatorDesign.pdf (accessed on 2 February 2024).