Engineering Applications of Artificial Intelligence 52 (2016) 1–9

Contents lists available at ScienceDirect

Engineering Applications of Artificial Intelligence

journal homepage: www.elsevier.com/locate/engappai

Nonlinear control of a boost converter using a robust regression based reinforcement learning algorithm

D. John Pradeep, Mathew Mithra Noel *, N. Arun

School of Electronics Engineering, VIT University, India

ARTICLE INFO

Article history:
Received 22 August 2015
Received in revised form
14 January 2016
Accepted 10 February 2016

Keywords:
Reinforcement learning
Boost converter
Non-linear control
Robust regression
Markov Decision Process

ABSTRACT

In this paper a reinforcement learning based nonlinear control strategy for control of boost converters is presented. Control of boost converters is a challenging nonlinear control problem, and classical linear control techniques perform poorly since the model of the converter depends on the state of the switching elements. In this paper the boost converter control problem is formulated as an optimal multi-step decision problem aimed at attaining a constant output voltage. Optimal multi-step decision problems can be solved using the framework of Markov Decision Processes (MDP) and Reinforcement Learning (RL); however iterative solution procedures exist only for discrete state problems. In this paper two possible approaches for applying RL to the boost converter problem are proposed. First a RL based control strategy for a discretized model of the boost converter problem is presented. Next an approach that applies robust regression to mitigate the effects of discretization by smoothly interpolating between the control decisions computed for the discretized states is presented. Simulation results indicate that the robust regression based RL strategy significantly reduces oscillations and overshoot and gives a better output voltage compared to the pure RL strategy.

1. Introduction

A boost converter (Sundareswaran and Sreedevi, 2009; Rashid, 2004) is a step-up DC to DC converter that finds extensive application in solar power, fuel cell, hybrid electric vehicle, LED, fluorescent lighting and battery technologies. Boost converters are primarily used to avoid the stacking of DC voltage sources in series to achieve higher voltages. All boost converters require a minimum of two switching devices and at least one energy storing element. Control of the output voltage of the boost converter is a challenging nonlinear control problem because the model of the converter depends on the states of the switching elements. Conventional controller design strategies using approximate linear models to control boost converters do not perform well, so exploration of alternate optimal and nonlinear strategies is of interest. This paper presents a machine learning based strategy for the solution of the boost converter control problem.

The boost converter control problem can be formulated as a sequential optimal decision problem if the framework of Markov Decision Processes (MDP) is adopted. This is advantageous since effective RL (Sutton and Barto, 1998; Kaelbling et al., 1996) based algorithms can be used to compute optimal control actions for MDP based models.

1.1. Reinforcement learning

RL is a branch of machine learning (Watkins and Dayan, 1992; Bertsekas and Tsitsiklis, 1996; Mitchell, 1997) that mimics the behaviour of an intelligent agent that learns to accomplish a task by choosing actions to maximize environmental rewards. The rewards can depend on just the state, or on both the state and action taken in the state. Use of RL in designing controllers for nonlinear control problems is reported in (Noel and Pandian, 2014; Fernandez-Gauna et al., 2014). RL was used in applications like game playing (Tesauro, 1994, 1992), controlling autonomous robots and scheduling (Ng et al., 2004; Shokri, 2011; Pazis and Lagoudakis, 2011; Wiering et al., 2011) and in industrial process control (Lewis and Vamvoudakis, 2011; Syafiie et al., 2011). The application of RL to practical problems is hindered by the 'curse of dimensionality' (Bellman, 1957), where the computational complexity increases exponentially with increase in number of discretization levels of the state space. In this paper a robust regression based function approximation strategy is used to mitigate the effects of discretization of the continuous state space.

In the RL learning paradigm the number of all possible states and actions is assumed to be finite. When the system is in state $\boldsymbol{s}$, the agent takes an action $\boldsymbol{a}$, that drives the system to the next state

* Corresponding author. Tel.: +91 9489343787.
E-mail addresses: johnpradeepdarsy@gmail.com (D.J. Pradeep), mathew.mithra@gmail.com (M.M. Noel), narun1929@gmail.com (N. Arun).

http://dx.doi.org/10.1016/j.engappai.2016.02.007
0952-1976/

$\boldsymbol{s}'$. The state $\boldsymbol{s}$ and the action $\boldsymbol{a}$ are in general vectors of real numbers. The agent receives a reward $R(\boldsymbol{s},\boldsymbol{a})$ from the environment that indicates the desirability of taking an action $\boldsymbol{a}$ in state $\boldsymbol{s}$. The goal of the agent is to maximize the expected value of cumulative discounted rewards by taking appropriate actions over time and this model is referred to as MDP. In this paper, discounted rewards are used to encourage the agent to achieve the goal state faster and to ensure a finite total reward. The extent to which future rewards are discounted can be controlled by changing the discount factor γ.

An MDP is thus characterized by a 5-tuple $(S, A, \gamma, P_{\boldsymbol{sa}}, R)$, where S is the set of all possible states, A is the set of all possible actions, γ is the discount factor, $P_{\boldsymbol{sa}}(\boldsymbol{s}')$ are the state transition probabilities and R is the reward function. In general the policy function π maps states to actions, $(\pi{:}S\rightarrow A)$ and the reward function R maps state action pairs to real numbers $(R{:}S\times A\rightarrow\mathbb{R})$. In some applications rewards do not depend on the action taken $(R{:}S\rightarrow\mathbb{R})$.

The value function V is the expected sum of discounted rewards for a given initial state and predetermined policy. If a policy π is being executed, when the system is in state $\boldsymbol{s}$, the action $\boldsymbol{a}$ is taken according to the policy indicated by $\boldsymbol{a}=\pi(\boldsymbol{s})$. The value function assigns a real number to each state that indicates the desirability of that state. The concept of a value function is a fundamental feature of the RL paradigm. Frequently the value function is easier to compute than the policy function. So, the policy function is computed from the value function in RL. The value function is defined by Eq. (1).

$$V^{\pi}(\boldsymbol{s}) = \mathrm{E}(R(\boldsymbol{s}_0)+\gamma R(\boldsymbol{s}_1)+\gamma^2 R(\boldsymbol{s}_2)+\cdots \mid \boldsymbol{s}_0=\boldsymbol{s},\pi) \tag{1}$$

In Eq. (1), discount factor $\gamma\in[0,1]$ helps in emphasizing present rewards and discounting future rewards. The goal of RL is to provide a best policy that maximizes the total discounted rewards. The optimal value function is the value function when the optimal policy is followed and is given by Eq. (2)

$$V^{*}(\boldsymbol{s}) = \max_{\pi} V^{\pi}(\boldsymbol{s}) \tag{2}$$

Bellman's equation for the optimal value function is given in Eq. (3) and it states that the expected cumulative discounted rewards obtained when starting in state $\boldsymbol{s}$ and following the optimal policy is equal to sum of the immediate reward $R(\boldsymbol{s})$ received for being in state $\boldsymbol{s}$ and the discounted maximum expected rewards from the next state $\boldsymbol{s}'$. This represents the stochastic case when transition to the next state is probabilistic.

$$V^{*}(\boldsymbol{s}) = R(\boldsymbol{s})+\gamma\max_{a\in A}\sum_{s'\in S} P_{\boldsymbol{sa}}(\boldsymbol{s}')V^{*}(\boldsymbol{s}') \tag{3}$$

In case of a deterministic system, all state transition probabilities are zero except for one state transition (for which the probability is 1). For the deterministic case Bellman's equation for the optimal value function given in Eq. (3) reduces to Eq. (4)

$$V^{*}(\boldsymbol{s}) = R(\boldsymbol{s})+\gamma\max_{a\in A} V^{*}(\boldsymbol{s}') \tag{4}$$

Any policy that maximizes the future discounted rewards is referred to as an optimal policy and is denoted by π^{*}. The optimal policy can be computed from the optimal value function with Eq. (5) which states that, the best action to take in state $\boldsymbol{s}$ is the action that maximizes the expected cumulative discounted rewards from the next state $\boldsymbol{s}'$.

$$\pi^{*}(\boldsymbol{s}) = \operatorname*{argmax}_{a\in A}\sum_{s'\in S} P_{\boldsymbol{sa}}(\boldsymbol{s}')V^{*}(\boldsymbol{s}') \tag{5}$$

If the system is deterministic, then the above equation reduces to Eq. (6)

$$\pi^{*}(\boldsymbol{s}) = \operatorname*{argmax}_{a\in A} V^{*}(\boldsymbol{s}') \tag{6}$$

Table 1
Nomenclature used in the formulation of the boost converter control problem.

S. no.	Symbol	Description
1	$\boldsymbol{s}$	system state vector
2	$\boldsymbol{a}$	control action vector or input to the system
3	$P_{\boldsymbol{sa}}(\boldsymbol{s}')$	state transition probabilities
4	$R(\boldsymbol{s},\boldsymbol{a})$	reward for taking action $\boldsymbol{a}$ in state $\boldsymbol{s}$
5	$\pi(\boldsymbol{s})$	action taken in state $\boldsymbol{s}$ following a policy π
6	$V^{\pi}(\boldsymbol{s})$	cumulative sum of discounted rewards for following policy π, starting from state $\boldsymbol{s}$
7	π^{*}	optimal policy function
8	V^{*}	optimal value function
9	$\boldsymbol{x}$	$[x_1\ x_2]^{\mathrm{T}}$ state vector of the boost converter system
10	D	set of all possible duty cycle values
11	γ	discount factor to favour immediate rewards
12	N_i	number of discretization levels for the state variable x_i
13	N_D	number of discretization levels for the duty cycle

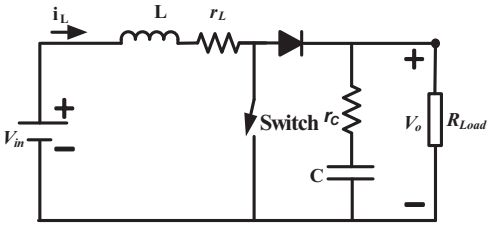

Fig. 1. Boost converter in open loop.

The nomenclature of variables used in this paper and their description is given in in Table 1.

1.2. Boost converter

A boost converter in open loop without feedback control is shown in Fig. 1. The behaviour of the boost converter can be modelled with two linear state space models; one model describes the boost converter system when the converter switch is ON and another model describes the system when the switch is OFF.

When the converter switch is ON, the inductor stores energy in its magnetic field and when the switch is OFF, the magnetic field is de-energized to maintain current flow to the load. The voltage seen at the load is the sum of input voltage and the voltage across the inductor aiding in achieving a higher output voltage. A boost converter in open loop does not provide good dynamic response and regulation characteristics, so it is always used in closed loop. A boost converter in a closed loop is shown in Fig. 2. The controller senses the present state of the boost converter and changes the duty cycle of the pulse width modulator to maintain a constant output voltage.

The use of linear control techniques like Proportional Integral and Derivative (PID) controllers for boost converter control is widely reported in literature. Traditional controller design methods (Hung et al., 1993; Cominos and Munro, 2002; Guo et al., 2003; Balestrino et al., 2006) aim at proper tuning of the proportional, integral and derivative constants so that the boost converter provides a constant output voltage. However linear control techniques described in current literature do not provide satisfactory performance due to the hard nonlinearity of the boost converter system. Perry et al. (2004) describe a PI like fuzzy controller while Sreekumar and Agarwal (2008) proposed a hybrid algorithm for voltage regulation in boost converters.

The organization of this paper is as follows: first the boost converter control problem is formulated as an optimal sequential decision problem (MDP), second a scheme that uses robust regression for effective solution using RL is presented, finally

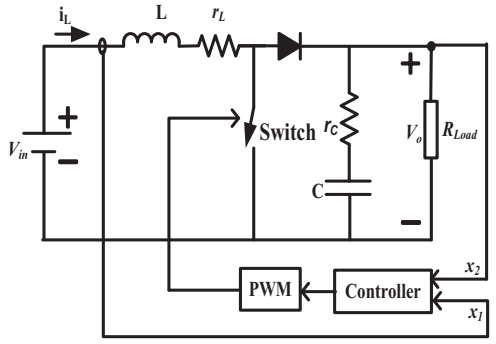

Fig. 2. Boost converter in closed loop.

results comparing the pure RL approach and the robust regression based RL approach proposed in this paper are presented.

2. Problem formulation

In the following, the boost converter control problem is modelled as an MDP consisting described by the 5-tuple $(X, D, P_{xd}, \gamma, R)$.

The set $X=\{[x_1(m), x_2(n)]: m=1 \text{ to } N_1 \text{ and } n=1 \text{ to } N_2\}$ where, x_1 is inductor current and x_2 is the capacitor voltage discretized to N_1 states and N_2 states respectively. $D=\{d(n): n=1 \text{ to } N_D\}$ is the set of duty cycle values given as input to the pulse width modulator. The duty cycle values are discretized to N_D states.

The probability of going to state $\boldsymbol{x}'$ when action d is taken in state $\boldsymbol{x}$ is denoted by $P_{\boldsymbol{xd}}(\boldsymbol{x}')$. Thus $P_{\boldsymbol{xd}}$ represent the state transitions probabilities. Since the boost converter is a deterministic system, $P_{\boldsymbol{xd}}$ is 0 except for one state transition. The next state $\boldsymbol{x}' \in X$ to which the system makes a transition depends only on the current state $\boldsymbol{x} \in X$ and action $d \in D$.

The state space model for the boost converter with the switching element in the ON state is given by Eq. (7) and by Eq. (8):

$$\begin{bmatrix}\dot{x}_1\\ \dot{x}_2\end{bmatrix}=\begin{bmatrix}\frac{-r_L}{L} & 0\\ 0 & \frac{-1}{C(R_{Load}+r_C)}\end{bmatrix}\begin{bmatrix}x_1\\ x_2\end{bmatrix}+\begin{bmatrix}\frac{1}{L}\\ 0\end{bmatrix}V_{in} \tag{7}$$

$$V_O(\boldsymbol{x})=\begin{bmatrix}0 & \frac{R_{Load}}{(R_{Load}+r_C)}\end{bmatrix}\begin{bmatrix}x_1\\ x_2\end{bmatrix} \tag{8}$$

The state space model for the boost converter with the switching element in the OFF state is given by Eq. (9) and Eq. (10):

$$\begin{bmatrix}\dot{x}_1\\ \dot{x}_2\end{bmatrix}=\begin{bmatrix}\frac{-r_L+R_{Load}//r_C}{L} & \frac{-R_{Load}}{L(R_{Load}+r_C)}\\ \frac{-R_{Load}}{C(R_{Load}+r_C)} & \frac{-1}{C(R_{Load}+r_C)}\end{bmatrix}\begin{bmatrix}x_1\\ x_2\end{bmatrix}+\begin{bmatrix}\frac{1}{L}\\ 0\end{bmatrix}V_{in} \tag{9}$$

$$V_O(\boldsymbol{x})=\begin{bmatrix}R_{Load}//r_C & \frac{R_{Load}}{(R_{Load}+r_C)}\end{bmatrix}\begin{bmatrix}x_1\\ x_2\end{bmatrix} \tag{10}$$

where V_{in} is the input voltage, L is the inductance value of the inductor, r_L is the equivalent resistance of the inductor, R_{Load} is the load resistance, C is the capacitance of the capacitor and r_C is the equivalent resistance of the capacitor. This switch in Fig. 1 is usually implemented with a metal–oxide–semiconductor field-effect transistor (MOSFET) in the physical realization of the boost converter system.

Eqs. (7)–(10) are linear differential equations when considered separately. However the linear state space model given by Eq. (7) and Eq. (8) describes the system when the switch in Fig. 1 is ON and Eq. (9) and Eq. (10) describe the system when the switch in Fig. 1 is OFF. Since the linear model depends on the state of the switching element the overall system is nonlinear. A linear system is by definition a system described by a single state space equation of the form $\frac{dx}{dt}=f(x,u)=Ax+Bu$, where the function f is linear. However in this case the function f is not described by a single linear equation and hence the overall boost converter system is nonlinear.

When the diode is conducting it has a small voltage of about 0.7 V across it and hence can be considered as a short circuit (ideal switch approximation) in power electronic applications. When the diode is not conducting it can be treated as an open circuit since its reverse leakage current is in microamperes. Thus the diode is treated as either a short circuit when it is forward biased (small forward voltage of 0.7 V is ignored) and as an open circuit when it is reverse biased (small reverse leakage current is ignored) in the diode model used in power electronic applications. Similarly the MOSFET switch is also treated as a short circuit when it is conducting and as an open circuit when it is not conducting. The conducting state of the MOSFET is controlled with a high or low gate control signal. Since both the diode and MOSFET are treated as open or short circuits depending on their state (conducting or not conducting), they do not explicitly appear in the state space model.

However it is worth noting that since the MOSFET is treated as a short circuit when conducting and as an open circuit when not conducting we get two different linear differential equations (Eqs. (7) and (8)) depending on the conducting state of the MOSFET. This results in the overall nonlinearity of the boost converter system.

In Fig. 1 when the MOSFET is switched ON with a high gate control signal all the source current flows through the MOSFET switch. No current flows through the diode as the diode is reverse biased by the capacitor voltage. In the conducting state the MOSFET behaves likes a near ideal switch and has a very small voltage across it while in the non-conducting state it behaves essentially like an open circuit. Applying KVL when the MOSFET is conducting results in Eq. (7). When the MOSFET is switched OFF with a low gate control signal all the inductor current is forced through the forward biased diode and capacitor. Neglecting the ON state small voltage of 0.7 V and applying KVL results in Eq. (9) in this case. The diode prevents the capacitor from discharging when the MOSFET switch is not conducting.

If the system is assumed to start in state $\boldsymbol{x}(0)=[x_1(0)\, x_2(0)]^T$ and the controller takes an action $d(0) \in D$, the system is driven into the new state $\boldsymbol{x}(1)=[x_1(1)\, x_2(1)]^T$ and in this new state an action $d(1) \in D$ is taken by the controller and the system is transitions to the next state. This process continues indefinitely as shown in Fig. 3.

RL aims at maximizing the total payoff by choosing the best action in every state. The value function $V^{\pi}(\mathbf{x})$ is the expected sum of discounted rewards as given in Eq. (11) and measures the desirability of the system being in state $\boldsymbol{s}$.

$$V^{\pi}(\boldsymbol{x})=E[(R(\boldsymbol{x}(0))+\gamma R(\boldsymbol{x}(1))+\gamma^2 R(\boldsymbol{x}(2))+\cdots|\boldsymbol{x}(0)=\boldsymbol{x},\pi] \tag{11}$$

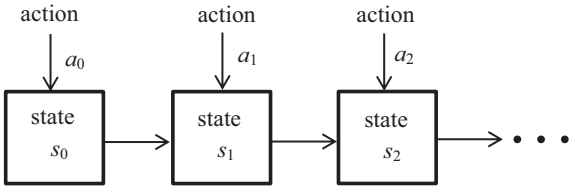

Fig. 3. Representation of the MDP for boost converter control problem with state transitions and control actions.

For a deterministic system, Eq. (11) reduces to Eq. (12) since the expectation of a constant random variable is the same constant.

$$V(\boldsymbol{x}) = R(\boldsymbol{x}(0)) + \gamma R(\boldsymbol{x}(1)) + \gamma^2 R(\boldsymbol{x}(2)) + \cdots \tag{12}$$

Eq. (12) can be used to obtain a recursive representation of the value function and is referred to as Bellman's equation, given in Eq. (13), using which the optimal value functions for all the states of the system are computed.

$$V^{\pi}(\boldsymbol{x}) = R(\boldsymbol{x}) + \gamma V^{\pi}(\boldsymbol{x}') \tag{13}$$

The first term in Eq. (13) represents the immediate reward the controller gets for being in state $\boldsymbol{x}$ and the second term represents the sum of future discounted rewards. $\gamma \in [0,1]$ is the discount factor which is used to favour actions leading to immediate rewards over actions leading to delayed rewards. The effect of variation in control performance with γ is explored in Section 4. When the controller adopts the optimal policy, the optimal value function is computed using Eq. (14)

$$V^{*}(\boldsymbol{x}) = \max_{\pi} V^{\pi}(\boldsymbol{x}) = R(\boldsymbol{x}) + \gamma \max_{d \in D} V^{*}(\boldsymbol{x}') \tag{14}$$

The optimal policy for boost converter problem is computed using Eq. (15)

$$\pi^{*}(\boldsymbol{x}) = \operatorname{argmax}_{d \in D} V^{*}(\boldsymbol{x}') \tag{15}$$

Value iteration and policy iteration are the two algorithms used to find the optimal value and best policy in RL. In this paper, value iteration algorithm is used to learn the optimal policy.

The reward function R in the formulation presented above represents the desirability of taking a specific action in a specific state. In this paper two different possible reward functions were considered. In the first reward function considered, voltages lower than the desired output voltage, are preferred to higher output voltages. So, the states that result in a higher output voltage are penalised compared to other states. The reward function for the above mentioned case is modelled as shown in Eq. (16).

$$R_1(\boldsymbol{x}) = \begin{cases} -k_1 \left| V_O^{desired} - V_O(\boldsymbol{x}) \right| & \text{for } V_O(\boldsymbol{x}) \geq V_O^{desired} \\ -k_2 \left| V_O^{desired} - V_O(\boldsymbol{x}) \right| & \text{for } V_O(\boldsymbol{x}) < V_O^{desired} \end{cases} \tag{16}$$

where k_1 and k_2 are positive constants and $k_1 > k_2$, $V_O(\boldsymbol{x})$ is the present output voltage and $V_O{}^{desired}$ is the desired output voltage.

For the second reward function Eq. (17), the reward is proportional to the difference between the current state and the desired state of the boost converter:

$$R_2(\boldsymbol{x}) = -k_3 \left| V_O^{desired} - V_O(\boldsymbol{x}) \right| \tag{17}$$

where k_3 is a positive constant.

3. Robust regression based reinforcement learning

Most control applications involve continuous state space and action space, whereas RL computes actions only for discrete states. Thus the continuous state and action spaces have to be discretized to apply RL. The discrete state space policy computed by RL, when used on a continuous state space model, results in oscillations and overshoots. Thus an effective function approximation scheme is needed to estimate the policy function for continuous state spaces and for this purpose least squares regression is frequently used. Use of least squares approach for finding the regression coefficients involves minimizing the sum of square residuals assuming the errors to be having finite variance and are uncorrelated with the regressors which is not true in many cases. Classical regression methods fail if there are outliers in the data, so robust regression (Huber, 1964; Street et al., 1988) methods such as M-estimators, Least Trimmed Squares and weighted least squares are of interest.

The squared residuals in least squares estimation are replaced by another function of residuals which is often termed as objective function for estimation of regression coefficients in robust regression. In the boost converter problem, the estimation of duty cycle for new states is modelled as an M-estimation problem (Fox, 2002). The duty cycle value (control action) for each state is written as a linear function of state variables (Eq. (18)) and is represented compactly using vector notation in Eq. (19):

$$d_i = \alpha + \beta_1 x_{1i} + \beta_2 x_{2i} + \varepsilon_i \tag{18}$$

$$d_i = \mathrm{X}_i'\beta + \epsilon_i \tag{19}$$

where, X_i' is the matrix representing state variables, β is matrix representing the estimation coefficients and ε_i is the error for i th observation and $d_i \in D$. The M-estimator minimizes the objective function ρ given in Eq. (20).

$$\sum_{i=1}^{n} \rho(\varepsilon_i) = \sum_{i=1}^{n} \rho(d_i - \mathrm{X}_i'\beta) \tag{20}$$

The objective function used in this paper is the Tukey-bisquare function given in Eq. (21):

$$\rho(\varepsilon) = \begin{cases} \frac{k^2}{6}\left\{1 - \left[1 - \left[\frac{\varepsilon}{k}\right]^2\right]^3\right\} & \text{for} |\varepsilon| \leq k \\ \frac{k^2}{6} & \text{for} |\varepsilon| > k \end{cases} \tag{21}$$

where k is the tuning constant of the objective function.

Eq. (20) can be rewritten as Eq. (22). Eq. (22) represents a weighted least squares problem and can be solved iteratively using the weighted least squares algorithm:

$$\sum_{i=1}^{n} w_i (b_i - \mathrm{X}_i'\beta)\mathrm{X}_i' = \mathbf{0} \tag{22}$$

where $w_i = w(\varepsilon_i)$ and $w(\varepsilon) = \psi(\varepsilon)/\varepsilon$ and $\psi = \rho'$.

The coefficients α, β_1 and β_2 computed are used to estimate the value of duty cycle as a linear function of state variables. In this work two possible robust regression based RL strategies were explored. In the first approach, robust regression is applied to the policy function directly-policy regression based RL (PRRL) algorithm and in the second approach, robust regression was applied to the value function-value regression based RL (VRRL) algorithm. The duty cycle values computed by PRRL algorithm may have values above 0.9 and below 0.1 which are not feasible in the boost converter problem considered. So the duty cycle values above 0.9 and below 0.1 are rounded off to 0.9 and 0.1 respectively. PRRL and VRRL algorithms are given in Table 2 and Table 3 respectively. In VRRL approach the control policy function for the continuous state space is computed from the value function learned over the discretized state space using robust regression.

4. Results

A benchmark boost converter system with circuit parameters given in Sundareswaran and Sreedevi (2009) was considered. The values of the circuit parameters are $V_{in} = 36$ V, $L = 33$ mH, $r_L = 3\ \Omega$, $R_{Load} = 100\ \Omega$, $C = 1000\ \mu$F, $r_C = 0.5\ \Omega$, $f = 2$ kHz and $V_O{}^{desired} = 80$ V. Here f is the frequency of the PWM signal. In this work the state variables x_1 and x_2 were discretized into 10 and 1000 levels respectively ($N_1 = 10$ and $N_2 = 1000$) and the duty cycle was discretized into 10 levels ($N_D = 10$). The range of values considered for the state variables and actions are: $x_1 \in [0,5]$, $x_2 \in [0,100]$ and $d \in [0.1,0.9]$ respectively. The diode and switch shown in Fig. 2 are

Table 2
Policy regression RL (PRRL) algorithm.

(1) Define $R(\boldsymbol{x})$, γ, and $V_O^{desired}$.
(2) Discretize x_1, x_2 and d into N_1, N_2 and N_D states respectively.
(3) For each state $\boldsymbol{x}$, initialize $V(\boldsymbol{x}):=0$.
(4) Repeat for N iterations
  Repeat for every state
  $V(\boldsymbol{x}) := R(\boldsymbol{x}) + \gamma \max_{d \in D} V^*(\boldsymbol{x}')$
  End
 End
(5) Repeat for each state
  $\pi^*(\boldsymbol{x}) := \operatorname{argmax}_{d \in D} V^*(\boldsymbol{x}')$
 End
(6) Use $\boldsymbol{x}$ and $\pi^*(\boldsymbol{x})$ to calculate the coefficients α, β_1 and β_2 using robust regression analysis with Tukey-bisquare function as the objective function. Let $\tilde{\pi}^*(\boldsymbol{x})$ computed using the regression coefficients be the approximation to $\pi^*(\boldsymbol{x})$.
(7) For continuous control, do the following
  (i) Sense the current state $\boldsymbol{x}_{\text{current}}$
  (ii) Compute $\tilde{d}(\boldsymbol{x}_{\text{current}}) = \tilde{\pi}^*(\boldsymbol{x}_{\text{current}})$
  (iii) Set the duty cycle value to $\tilde{d}(\boldsymbol{x}_{\text{current}})$
  (iv) Go to (i)

Table 3
Value regression RL (VRRL) algorithm.

(1) Define $R(\boldsymbol{x})$, γ, and $V_O^{desired}$.
(2) Discretize x_1, x_2 and d into N_1, N_2 and N_D states respectively.
(3) For each state $\boldsymbol{x}$, initialize $V(\boldsymbol{x}):=0$.
(4) Repeat for N iterations:
  Repeat for every state:
  $V(\boldsymbol{x}) := R(\boldsymbol{x}) + \gamma \max_{d \in D} V^*(\boldsymbol{x}')$
  End
 End
(5) Use $\boldsymbol{x}$ and $V^*(\boldsymbol{x})$ to find the coefficients α, β_1 and β_2 using robust regression. Let $\tilde{V}^*(\boldsymbol{x})$ computed using the regression coefficients be the approximation to $V^*(\boldsymbol{x})$.
(6) For continuous control, do the following
  (i) Sense the current state $\boldsymbol{x}_{\text{current}}$
  (ii) Compute $\tilde{d}(\boldsymbol{x}_{\text{current}}) = \tilde{\pi}^*(\boldsymbol{x}_{\text{current}}) = \operatorname{argmax}_{d \in D} \tilde{V}^*(\boldsymbol{x}')$
  (iii) Set the duty cycle value to $\tilde{d}(\boldsymbol{x}_{\text{current}})$
  (iv) Go to (i)

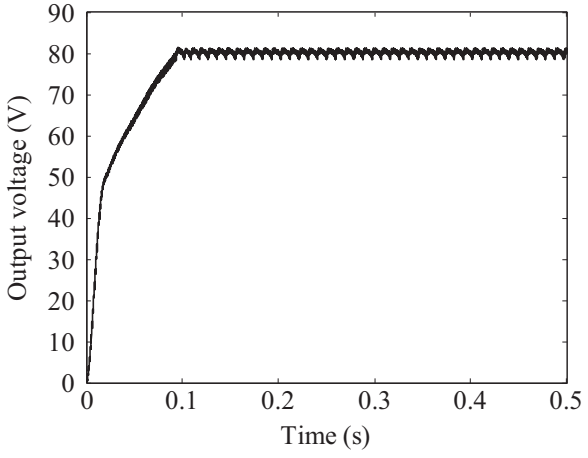

Fig. 4. Control of the boost converter output voltage with the pure RL strategy and reward function R_1.

considered to be ideal. The duration for the simulation was taken to be 500 ms.

The performance of pure the RL based control policy with reward function given in Eq. (16) is shown in Fig. 4 and the output voltage in steady state is shown in Fig. 5. The values of k_1 and k_2

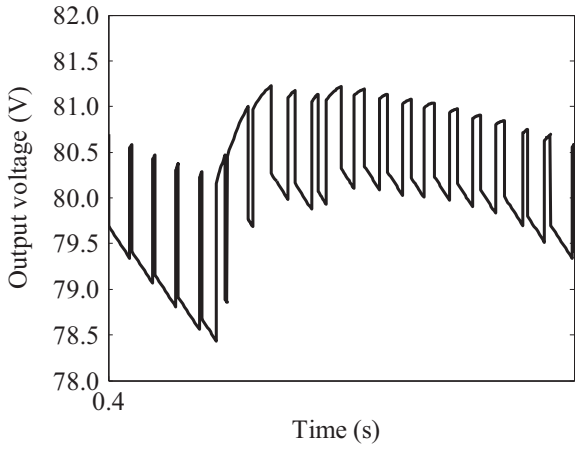

Fig. 5. Magnified view of the output voltage of the boost converter in steady state for the pure RL strategy and reward function R_1.

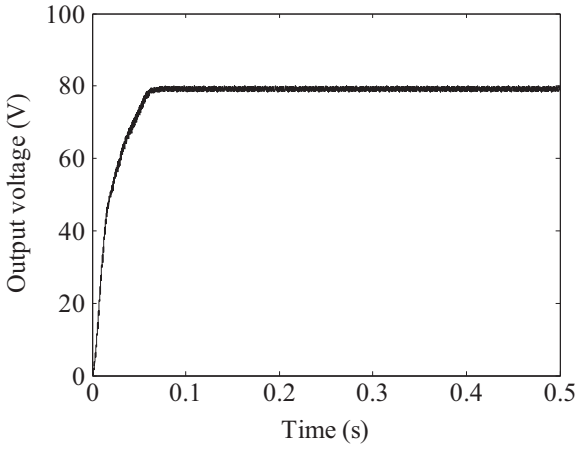

Fig. 6. Control of the boost converter output voltage with the pure RL strategy and reward function R_2.

were considered as 10 and 1 respectively in the reward function. The output voltage oscillates between 81.25 V and 78.5 V giving a non-uniform peak to peak ripple of 2.75 V with an average value of 80 V.

The performance of pure RL strategy with reward function Eq. (17) with k_3 as 100 is shown in Figs. 6 and 7. From Fig. 6, it can be observed that the output voltage has lower non-uniform ripple compared to Fig. 4 and that the steady state output voltage oscillates between 80.1 V and 78.4 V with a DC value of 79.5 V.

The robustness of the RL control strategy can be tested by observing the change in the output voltage when R_{Load} is changed from its nominal value of 100 Ω. Table 4 shows the percentage change in load voltage (voltage regulation) when the load resistance in varied. Since the voltage regulation obtained is within 2% the RL control strategy is robust to load variations.

The plot of the optimal value function with reward functions R_1 given in Eq. (16) and R_2 given in Eq. (17) are shown in Figs. 8 and 9 respectively. A comparative plot of duty cycle variation in the steady state, for a boost converter with reward functions R_1 and R_2 is given in Fig. 10. From Fig. 10, it can be observed that the optimal control policy with reward functions R_1 and R_2 are significantly different.

The effect of the choice of discount factor 'γ' on the control performance of boost converter is shown in Fig. 11. Simulation results indicate that when γ is chosen between 0.9 and 0.7, the output voltage converges to 79.5 V (a value near to the desired output voltage). As γ decreases from 0.9 to 0.7 an increase in rise

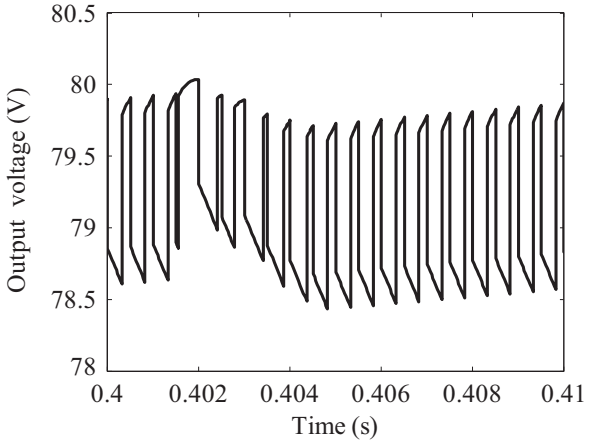

Fig. 7. Magnified view of the output voltage of the boost converter in steady state for the pure RL strategy and reward function R_2.

Table 4
Variation of the load voltage with load for the RL control strategy.

R_{Load} (Ω)	Change in R_{Load} (%)	Load regulation (%)
110	10	−0.15
120	20	−0.39
130	30	−0.73
140	40	−0.83
150	50	−0.93
160	60	−1.00
170	70	−1.05
180	80	−1.14
190	90	−1.15
200	100	−2.00

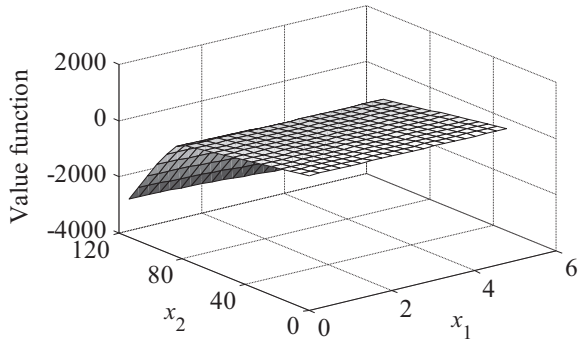

Fig. 8. Plot of the optimal value function versus the state variables for the pure RL strategy and reward functions R_1.

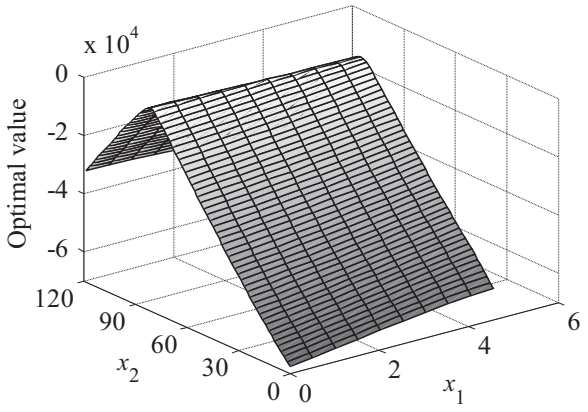

Fig. 9. Plot of the optimal value function versus the state variables for the pure RL strategy and reward functions R_2.

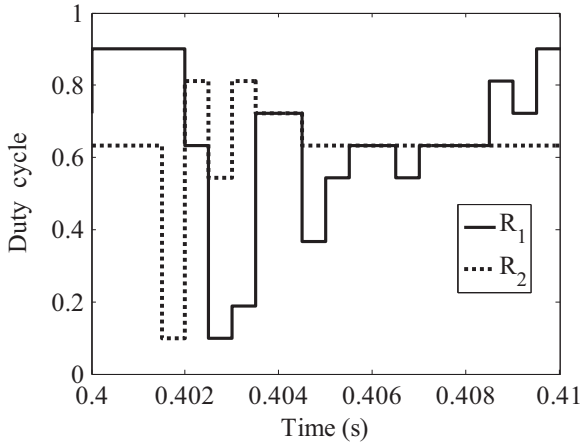

Fig. 10. Control actions taken by the pure RL controller in steady state for reward functions R_1 and R_2.

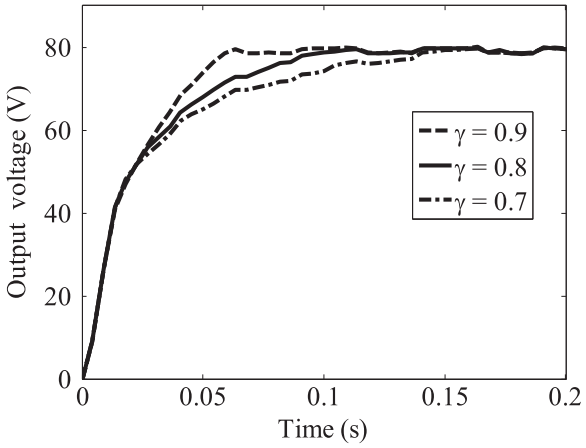

Fig. 11. Effect of variation of discount factor (gamma), on control performance with the pure RL strategy and reward function R_2.

time of the output voltage is observed. This degradation in performance is due to the higher weight assigned to immediate rewards compared to delayed rewards when γ is decreased as given by Eq. (12). Further decrease in γ to 0.6 and 0.5 made the output voltage settle at 70 V (a value much smaller than the desired output voltage of 80 V). The above results indicate that the choice of discount factor γ is critical in achieving good step response.

The output voltage of a boost converter using PRRL controller with reward functions R_1 and R_2 are shown in Figs. 12 and 14 respectively. This shows a robust response when compared to the response obtained using the pure RL based controller. The steady state output voltages in Figs. 13 and 15 show an oscillation of output voltage between 79.4 V and 80.6 V, with a peak to peak uniform ripple of 1.1 V and an average DC value of 80 V. A comparative plot of duty cycle in steady state for RL based controller and PRRL controller with reward functions $R_1$and R_2 are given in Figs. 16 and 17 respectively. Both the plots show that the duty cycle value for a regression based controller is constant in steady state. Plot of the optimal value function with robust regression based interpolation is shown in Fig. 18. The plot of output voltage of the boost converter with the VRRL controller is given in Fig. 19. This shows that the steady state output voltage has a DC value of 100 V which is not the desired output voltage. So the PRRL control strategy performs well when compared to the VRRL control strategy. Comparing Figs. 14 and 19 it can be seen that the PRRL strategy performs better than the VRRL strategy. The regression

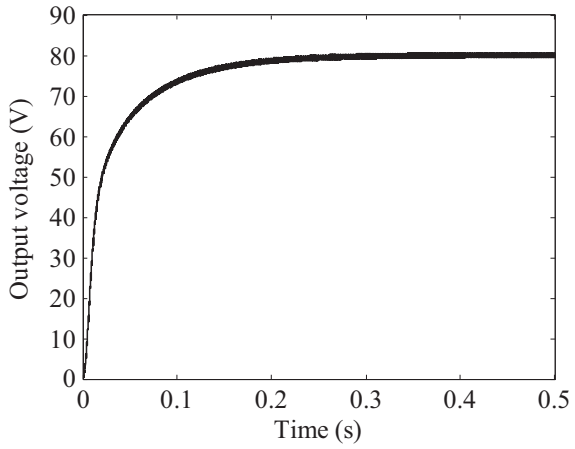

Fig. 12. Control of the boost converter output voltage with the PRRL strategy and reward function R_1.

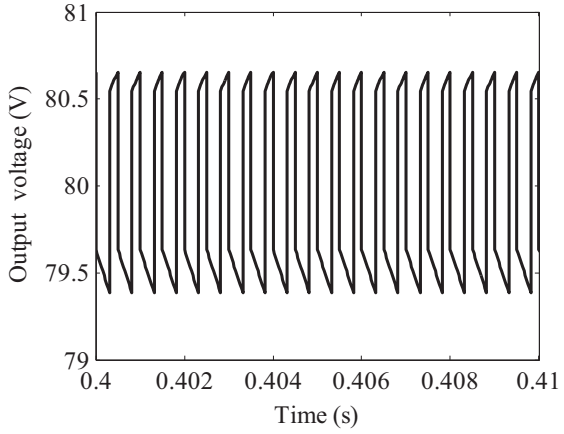

Fig. 13. Magnified view of the output voltage of the boost converter in steady state for the PRRL strategy and reward function R_1.

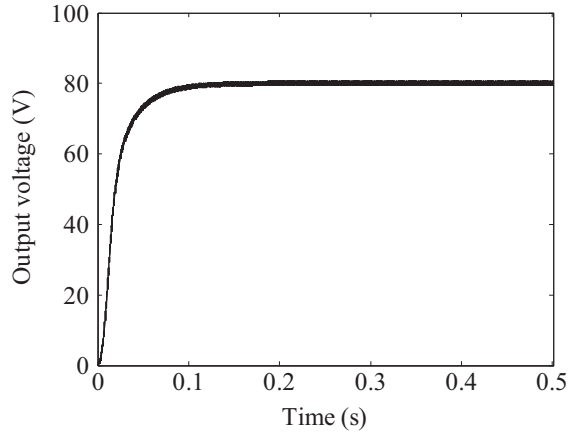

Fig. 14. Output voltage of a boost converter using PRRL based control policy with reward function R_2.

coefficients computed using VRRL algorithm are given in Table 5. The PRRL control strategy performs better as it attempts to learn the optimal control policy directly instead of indirectly from the value function as done by the VRRL strategy.

The parameter settings used to generate the results in this paper are presented in Table 5.

The plots of inductor current (state variable x_1) of a boost converter trained using reward function R_1 for an RL based

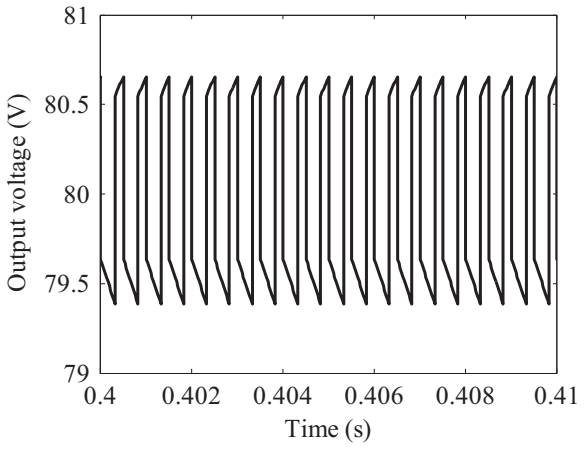

Fig. 15. Magnified output voltage of a boost converter in steady state using PRRL control policy with reward function R_2.

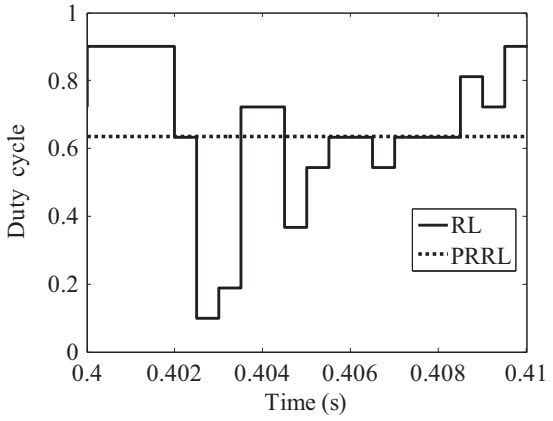

Fig. 16. Control actions taken by the pure RL and PRRL controllers in steady state for reward R_1.

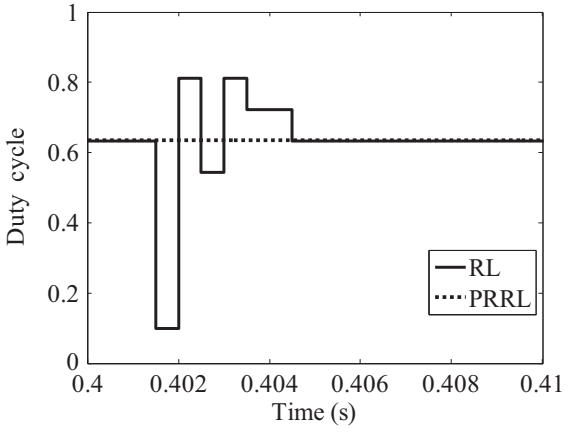

Fig. 17. Control actions taken by the pure RL and PRRL controllers in steady state for reward R_2.

controller and PRRL controller are given in Figs. 21 and 22 respectively. Fig. 20 shows a non-uniform charging and discharging pattern whereas Fig. 21 shows an even charging and discharging of the inductor. Thus the control policy learnt by a pure RL strategy is oscillatory. The control policy of the PRRL controller is a constant in steady state and results in a small uniform ripple in the output voltage as indicated in Fig. 13.

Best policy as a function of state variables for the pure RL and PRRL based control strategies are shown in Figs. 22 and 23

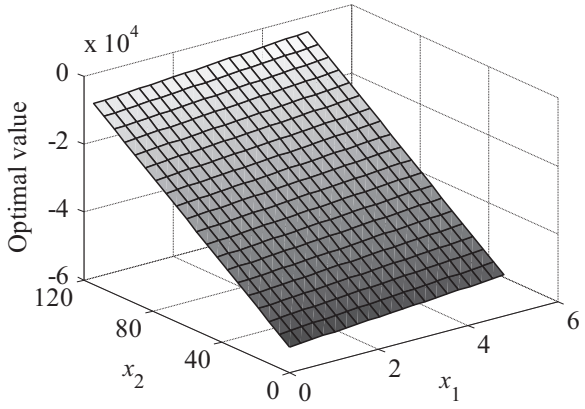

Fig. 18. Plot of the optimal value function versus the state variables for the VRRL strategy.

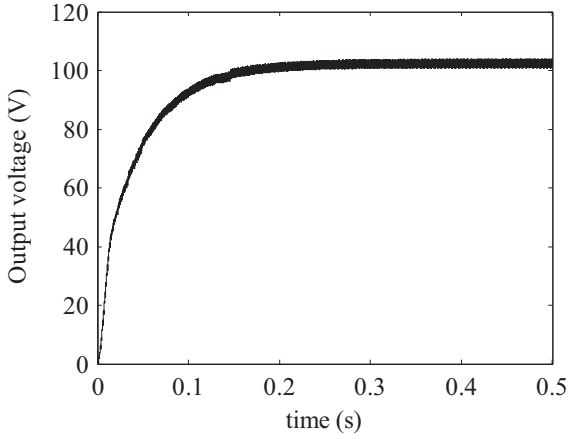

Fig. 19. Control of the boost converter output voltage with the VRRL strategy.

Table 5
Parameter settings for different algorithms.

Algorithm	k_1	k_2	k_3	α	β_1	β_2
PRRL (with R_1)	10	1	–	0.3265	−0.1191	0.0069
PRRL (with R_2)	–	–	100	0.9791	−0.0941	−0.0019
VRRL (with R_2)	–	–	100	−1.5457e5	4.4252e3	1.2202e3

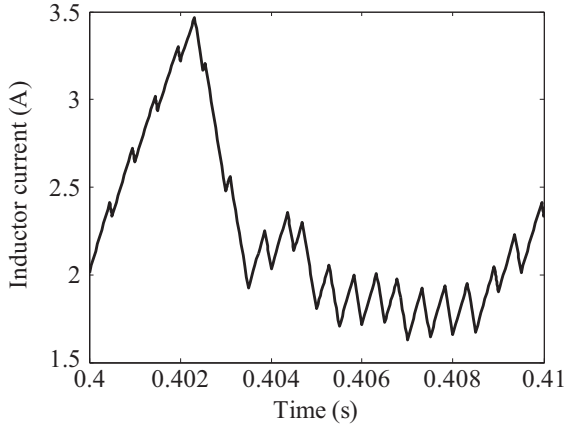

Fig. 20. Plot of inductor current (state variable x_1) of a boost converter in steady state using RL control policy.

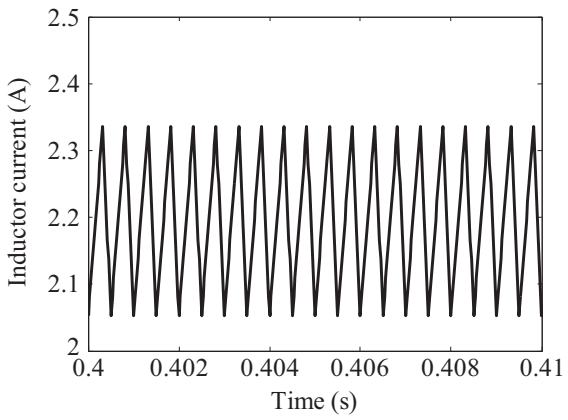

Fig. 21. Plot of inductor current (state variable x_1) of the boost converter in steady state using PRRL control policy.

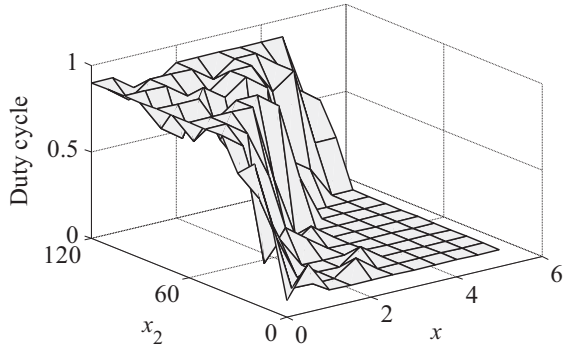

Fig. 22. Plot of the control action versus state variables for the pure RL based controller with reward function R_1.

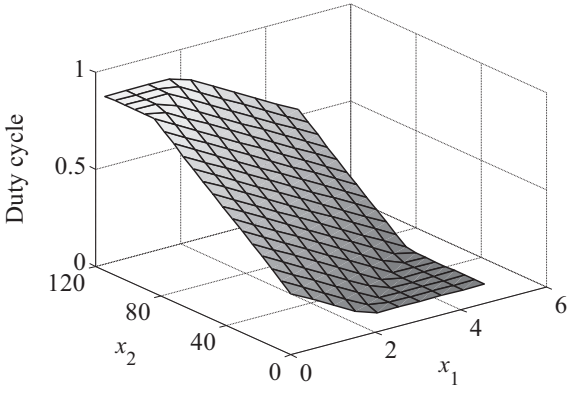

Fig. 23. Plot of the control action versus state variables for the policy regression based RL controller with reward function R_1.

respectively. The plot of best policy using PRRL algorithm results in a smoother surface which helps in continuous control of the boost converter. The control surface shown in Fig. 23 is piecewise linear but not linear. Any linear function in two state variables x_1 and x_2 must necessarily be of the form ax_1+bx_2 (or geometrically a plane) with a constant gradient. This is clearly not the case here since the gradient of the control surface is different in different regions (bends in the surface).

The control surface (Fig. 23) exhibits nonlinear saturation effects and constrains the control action (duty cycle) in the range [0,1] on the other hand a linear function is always unbounded and hence cannot constrain its output within a finite range. Any linear function must necessarily be unbounded since $L(c\boldsymbol{x})=cL(\boldsymbol{x})$ and c can be taken to be arbitrarily large. Also linear functions are

Table 6
Comparison of different control strategies.

Control Algorithm	% Overshoot	Rise time (ms)	Settling time (ms)
RL (R_1)	0.2536	0.0665	0.4975
RL (R_2)	0.4149	0.0424	0.0626
PRRL (R_1)	0	0.0835	0.2893
PRRL (R_2)	0	0.0392	0.1308
PID (Sundareswaran and Sreedevi, 2009)	0	35.6	0.1600

differentiable and have a well-defined gradient (first derivative) at each point. However the gradient of the control surface in Fig. 23 changes discontinuously (bends in the control surface) and hence the control policy is highly nonlinear and non-differentiable (no unique derivative at all points). Thus the control policy is highly nonlinear. Fig. 22 which shows the control surface for the pure RL approach is also highly irregular and nonlinear. A comparison of Figs. 22 and 23 show that the proposed robust regression approach results in a smoother nonlinear control surface.

A comparison of dynamic step response parameters for the RL based nonlinear control strategies presented in this paper and linear strategies from literature is presented in Table 6. Table 6 indicates that the performance of the PRRL boost converter control strategy is significantly better than other nonlinear RL based strategies as well as linear control strategies.

5. Conclusions

Control of power converters like boost converters is a challenging nonlinear control problem since the model of the converter depends on the states of the switching devices. Power converters are finding increasing application in key areas so the development of general nonlinear optimal control strategies for effective control of power converters is of interest. In this paper the boost converter control problem is reformulated as an optimal sequential decision and solved using the framework of MDP and RL. Two RL based strategies (PRRL and VRRL) that achieve optimal control of the nonlinear boost converter system were explored. The PRRL strategy which attempts to learn the optimal control policy directly performed better than the indirect VRRL strategy. The PRRL strategy overcomes the problem of oscillation and overshoot in the output voltage associated with linear and pure RL control strategies. Simulation results indicate PRRL strategy proposed in this paper is an effective approach for optimal control of boost converter systems. A limitation of the approach proposed in this paper is the dependence of this approach on the availability of an accurate state space model of the boost converter system. Accurate state space models can be easily constructed for low frequency converters however significant parasitic effects complicate the models of high frequency (30–300 MHz) converters. Thus for high frequency converter control applications alternate model-free RL approaches can be explored. Since the policy function exhibits sudden changes in value, the PRRL approach can possibly be improved by using wavelet based function approximation techniques which can model sharp discontinuities in the policy function. The application of the PRRL strategy proposed in this paper to a variety of more complex power converter control problems can also be considered for future work.

References

Balestrino, A., Corsanini, D., Landi, A., Sani, L., 2006. Circle-based criteria for performance evaluation of controlled DC–DC switching converters. IEEE Trans. Ind. Electron. 53. http://dx.doi.org/10.1109/TIE.2006.885157.

Bellman, R.E., 1957. Dynamic programming. Ann. Oper. Res. . http://dx.doi.org/10.1007/BF02188548

Bertsekas, D.P., Tsitsiklis, J.N., 1996. Neurodynamic Programming. Athena Scientific, USAhttp://dx.doi.org/10.1109/MCSE.1998.683749.

Cominos, P., Munro, N., 2002. PID controllers: recent tuning methods and design to specification. IEE Proc. Control Theory Appl. 149 (1), 46–53. http://dx.doi.org/10.1049/ip-cta:20020103.

Fernandez-Gauna, B., Ansoategui, I., Etxeberria-Agiriano, I., Graña, M., 2014. Reinforcement learning of ball screw feed drive controllers. Eng. Appl. Artif. Intell. 30, 107–117. http://dx.doi.org/10.1016/j.engappai.2014.01.015.

Fox, J., 2002. Nonparametric Regression. Appendix to An R and S-Plus Companion to Applied Regression. 2; , pp. 1–15. http://dx.doi.org/10.1016/0047-259X(91)90079-H.

Guo, L., Hung, J.Y., Nelms, R.M., 2003. Digital controller design for buck and boost converters using root locus techniques. In: Proceedings of the 29th Annual Conference of the IEEE Industrial Electronics Society, Vol. 2, IECON '03, (IEEE Cat. No.03CH37468). ⟨doi:10.1109/IECON.2003.1280344⟩.

Huber, P.J., 1964. Robust Estimation of a Location Parameter. Ann. Math. Stat. . http://dx.doi.org/10.1214/aoms/1177703732

Hung, J.Y., Gao, W., Hung, J.C., 1993. Variable structure control: a survey. IEEE Trans. Ind. Electron. 40, 2–22. http://dx.doi.org/10.1109/41.184817.

Kaelbling, L.P., Littman, M.L., Moore, A.W., 1996. Reinforcement learning: a survey. J. Artif. Intell. Res. 4, 237–285. http://dx.doi.org/10.1613/jair.301.

Lewis, F.L., Vamvoudakis, K.G., 2011. Reinforcement learning for partially observable dynamic processes: adaptive dynamic programming using measured output data. IEEE Trans. Syst. Man Cybern. Part B: Cybern. 41, 14–25. http://dx.doi.org/10.1109/TSMCB.2010.2043839.

Mitchell, T.M., 1997. Machine Learning, Annual Review of Computer Sciencehttp://dx.doi.org/10.1145/242224.242229.

Ng, A.Y., Kim, H.J., Jordan, M.I., Sastry, S., 2004. Inverted autonomous helicopter flight via reinforcement learning. In: Proceedings of the International Symposium on Experimental Robotics.

Noel, M.M., Pandian, B.J., 2014. Control of a nonlinear liquid level system using a new artificial neural network based reinforcement learning approach. Appl. Soft Comput. 23, 444–451. http://dx.doi.org/10.1016/j.asoc.2014.06.037.

Pazis, J., Lagoudakis, M.G., 2011. Reinforcement learning in multidimensional continuous action spaces, In: Proceedings of the IEEE Symposium on Adaptive Dynamic Programming and Reinforcement Learning, IEEE SSCI 2011: Symposium Series on Computational Intelligence-ADPRL 2011, pp. 97–104. ⟨doi:10.1109/ADPRL.2011.5967381⟩.

Perry, A.G., Feng, G., Liu, Y.F., Sen, P.C., 2004. A new design method for PI-like fuzzy logic controllers for DC-to-DC converters. In: Proceedings of the IEEE Annual Power Electronics Specialists Conference, PESC Record, pp. 3751–3757. ⟨doi:10.1109/PESC.2004.1355138⟩.

Rashid, M.H., 2004. Power Electronics: Circuits, Devices and Application.

Shokri, M., 2011. Knowledge of opposite actions for reinforcement learning. Appl. Soft Comput. J. 11, 4097–4109. http://dx.doi.org/10.1016/j.asoc.2011.01.045.

Sreekumar, C., Agarwal, V., 2008. A hybrid control algorithm for voltage regulation in DC–DC boost converter. IEEE Trans. Ind. Electron. 55 (6), 2530–2538. http://dx.doi.org/10.1109/TIE.2008.918640.

Street, J.O., Carroll, R.J., Ruppert, D., 1988. A note on computing robust regression estimates via iteratively reweighted least squares. Am. Stat. 42, 152–154. http://dx.doi.org/10.1080/00031305.1988.10475548.

Sundareswaran, K., Sreedevi, V.T., 2009. Boost converter controller design using queen-bee-assisted GA. IEEE Trans. Ind. Electron., 778–783. http://dx.doi.org/10.1109/TIE.2008.2006026.

Sutton, R.S., Barto, A.G., 1998. Reinforcement Learning : an Introduction. A Bradford Book.

Syafiie, S., Tadeo, F., Martinez, E., Alvarez, T., 2011. Model-free control based on reinforcement learning for a wastewater treatment problem. Appl. Soft Comput. . http://dx.doi.org/10.1016/j.asoc.2009.10.018

Tesauro, G., 1994. TD-Gammon, a self-teaching backgammon program, achieves master-level play. Neural Comput. . http://dx.doi.org/10.1162/neco.1994.6.2.215

Tesauro, G., 1992. Practical issues in temporal difference learning. Mach. Learn. 8, 257–277. http://dx.doi.org/10.1007/BF00992697.

Watkins, C.J.C.H., Dayan, P., 1992. Q-learning. Mach. Learn. 8, 279–292. http://dx.doi.org/10.1007/BF00992698.

Wiering, M.A., Hasselt, H. van, Pietersma, A.-D., Schomaker, L., 2011. Reinforcement learning algorithms for solving classification problems. In: Proceedings of the IEEE Symposium on Adaptive Dynamic Programming and Reinforcement Learning (ADPRL), pp. 91–96. ⟨doi:10.1109/ADPRL.2011.5967372⟩.