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
Available online 24 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

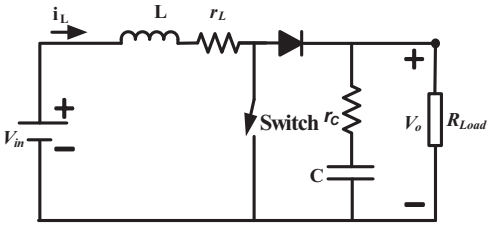

Fig. 1. Boost converter in open loop.

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