Applied Soft Computing 23 (2014) 444–451

Contents lists available at ScienceDirect

Applied Soft Computing

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

Control of a nonlinear liquid level system using a new artificial neural network based reinforcement learning approach

Mathew Mithra Noel, B. Jaganatha Pandian*

School of Electrical Engineering, VIT University, Vellore, Tamil Nadu 632014, India

ARTICLE INFO

Article history:
Received 1 February 2013
Received in revised form 3 February 2014
Accepted 20 June 2014
Available online 5 July 2014

Keywords:
Reinforcement learning
Neural networks
Nonlinear control
Process control

ABSTRACT

Most industrial processes exhibit inherent nonlinear characteristics. Hence, classical control strategies which use linearized models are not effective in achieving optimal control. In this paper an Artificial Neural Network (ANN) based reinforcement learning (RL) strategy is proposed for controlling a nonlinear interacting liquid level system. This ANN-RL control strategy takes advantage of the generalization, noise immunity and function approximation capabilities of the ANN and optimal decision making capabilities of the RL approach. Two different ANN-RL approaches for solving a generic nonlinear control problem are proposed and their performances are evaluated by applying them to two benchmark nonlinear liquid level control problems. Comparison of the ANN-RL approach is also made to a discretized state space based pure RL control strategy. Performance comparison on the benchmark nonlinear liquid level control problems indicate that the ANN-RL approach results in better control as evidenced by less oscillations, disturbance rejection and overshoot.

Introduction

Control of liquid level in multiple interacting tanks by adjusting flow rates is a paradigmatic nonlinear control problem that is ubiquitous in many industrial processes. Conventional control strategies like PID control that use approximate linear models do not perform well while undergoing large changes in the operating point. In this paper a machine learning [1–3] based approach that uses a new reinforcement learning strategy to achieve state regulation of a nonlinear system is proposed and applied to a benchmark nonlinear liquid level control problem.

Historically reinforcement learning (RL) has been applied in the fields of artificial intelligence and machine learning to solve optimal sequential decision making problems arising in game playing, scheduling and robotics [4–9]. Application of RL to enable autonomous agents to learn to make optimal decisions in real time is explored in [10–13]. Application of RL to a controller scheduling problem is considered in [14]. Application of RL strategies to tune ANN and fuzzy controllers is explored in [15–19]. Recently control of industrial processes using RL strategies was proposed [20–24].

Reinforcement learning algorithms solve the very general problem of optimal policy choice in a sequential decision making process. Consider a controller that attempts to control the state ‘*s*’ of a plant by taking actions ‘*a*’ that depend on the state. When the controller performs an action ‘*a*’ on the plant in state ‘*s*’, it receives a reward $R(s,a)$ that depends in general on both the action and the state. As a result of the action ‘*a*’ taken by the controller the controlled system transitions to the next state ‘*s*’ either probabilistically according to some probability distribution $P_{sa}(s')$ or deterministically according to some state transition law: $s' = f(s, a)$. The optimal policy $\pi^*(s)$ or action sequence is that which maximizes the expected cumulative discounted reward given by Eq. (1):

$$V^{\pi}(s) = E[R(s_0) + \gamma R(s_1) + \gamma^2 R(s_2) \cdots | s_0 = s, \pi] \qquad (1)$$

The expected cumulative discounted reward starting at some state s_0 is denoted by $V^{\pi}(s)$ since it depends on that initial state and also on the sequence of actions performed in each state $\pi(s)$. The constant γ in Eq. (1) is chosen in the set [0,1) to favor policies that provide immediate rewards. Also choice of γ ensures convergence of the infinite sequence in Eq. (1). This optimal policy choice problem is a Markov Decision Process (MDP) which is represented by a 5-tuple $(S, A, P_{sa}, \gamma, R)$ where:

S – finite set of states (discretization necessary to deal with continuous state spaces),

* Corresponding author. Tel.: +91 9840571695.
E-mail addresses: mathew.m@vit.ac.in (M.M. Noel), jaganathapandian@vit.ac.in, jaganathapandian@gmail.com (B.J. Pandian).

http://dx.doi.org/10.1016/j.asoc.2014.06.037
1568-4946/© 2014 Elsevier B.V. All rights reserved.

Nomenclature

s	system state vector
a	control action or input to the system
$P_{sa}(s')$	state transition probabilities
$R(s,a)$	reward for taking action 'a' in state 's'
$\pi(s)$	policy function or action to be taken in state 's'
$V^{\pi}(s)$	cumulative discounted reward for following policy π starting from state 's'
π^*	optimal policy function
$V^*(s)$	optimal value function
$\mathbf{h}$	$[h_1\ h_2\ h_3]^{\mathrm{T}}$, state vector for the liquid level system
Q	inlet flow rate for the liquid level system
γ	discount factor to discount future rewards and favor immediate rewards
N_i	number of discretization levels used for level variable h_i
N_f	number of discretization levels used for inlet flow rate Q

A – finite set of actions (discretization necessary to deal with continuous actions),
P_{sa} – probability distribution of the next state given the current state and action taken (includes deterministic transitions as a special case),
$\gamma \in [0, 1)$ – discount factor to discount rewards obtained in the future
$R : S \times A \to \mathbb{R}$ – reward function.

The policy π is a function $\pi : S \to A$ that maps the current state to the action to be taken by the controller. The optimal policy π^* is the policy that maximizes the total payoff:

$\pi * (s) = \max_{\pi} V^{\pi}(s)$. $V^{\pi}(s)$ is the expected cumulative discounted reward starting at some state s and following policy π and is known as the value function. The optimal value function is the value function obtained when the optimal policy is executed. The optimal value function satisfies Bellman equations:

$$V^*(s) = R(s) + \max_{a\in A} \gamma \sum_{s' \in S} P_{sa}(s') V^*(s') \tag{2}$$

The value iteration algorithm computes the optimal value function by iteratively using Eq. (2) starting with an estimate of all zeros. Once the optimal value function is known the optimal policy can be calculated from:

$$\pi^*(s) = arg\max_{a\in A} \sum_{s' \in S} P_{sa}(s') V^*(s') \tag{3}$$

The RL strategy given above works with finite state and action spaces so to apply RL to the liquid level system we discretize the states and actions. Simulation results indicate that the approach proposed in this paper that exploits the generalization ability of artificial neural networks (ANNs) to minimize the effect of state discretization results in less oscillations and overshoot of the liquid level.

Problem formation

The implementation of reinforcement learning begins with the definition of Markov Decision Process (MDP), which is a 5-tuple (H, Q_1, P_{hq}, γ, R) in the control problem considered.

Here, $H = \{[h_1(m)\, h_2(n)]^{\mathrm{T}} : m = 1 \text{ to } N_1 \text{ and } n = 1 \text{ to } N_2\}$, where, h_1 and h_2 are the liquid heights in tanks 1 and 2 respectively. The continuous heights are discretized into N_1 and N_2 levels. Thus the set of states H has size $N_1 \times N_2$.

$Q_1 = \{q_1(n) : n = 1 \text{ to } N_f\}$ is the set of all possible inlet flow rates to the first tank of the process. The inlet flow rate is taken as the action executed by the controller. This action is discretized into N_f levels.

P_{hq} – In the liquid level system the state transitions are deterministic so all probabilities except one are zero. A discretized version of the state space model of the system provided in Eq. (4) was used to find the next state $\mathbf{h}' \in H$ based on the current action $q_1 \in Q_1$ taken and the present state $\mathbf{h} \in H$.

$$\begin{aligned} \frac{dh_1}{dt} &= \frac{(q_1 - r_1\sqrt{h_1} - r_3\sqrt{h_1 - h_2})}{A_1} \\ \frac{dh_2}{dt} &= \frac{(q_2 - r_2\sqrt{h_2} + r_3\sqrt{h_1 - h_2})}{A_2} \end{aligned} \tag{4}$$

$\gamma \in [0, 1)$ – The discount factor used to give different weights to short term and long term rewards. γ was taken to be 0.99 in this paper.

R – Reward function, which rewards the controller for being in a state.

Possible reward functions for the control problem are given in Eqs. (5) and (6)

$$R(\mathbf{h}) = -C\left\|\mathbf{h}_{\text{desired}} - \mathbf{h}\right\| \tag{5}$$

$$R(\mathbf{h}) = \begin{cases} -1, & \text{if } \left\|\mathbf{h}_{\text{desired}} - \mathbf{h}\right\| \geq \delta \\ 0, & \text{otherwise} \end{cases} \tag{6}$$

where, $\mathbf{h}$ – present state; $\mathbf{h}_{\text{desired}}$ – desired state; $C > 0$ – a positive real number.

The system starts in some state $\mathbf{h(0)} = [h_1(0)\, h_2(0)]^{\mathrm{T}}$ and the controller takes some action, $q_1(0) \in Q_1$. This selected action takes the system to a new state $\mathbf{h(1)} = [h_1(1)\, h_2(1)]^{\mathrm{T}}$. From this state the controller takes the next action $q_1(1)$ and this process continues with successive action till the desired state is reached as given below.

$$\begin{bmatrix} h_1(0) \\ h_2(0) \end{bmatrix} \xrightarrow{q_1(0)} \begin{bmatrix} h_1(1) \\ h_2(1) \end{bmatrix} \xrightarrow{q_1(1)} \begin{bmatrix} h_1(2) \\ h_2(2) \end{bmatrix} \xrightarrow{q_1(2)} \cdots \tag{7}$$

Upon visiting the sequence of states $\mathbf{h(0)}$, $\mathbf{h(1)}$, $\mathbf{h(2)}$,... with actions $q_1(0)$, $q_1(1)$, $q_1(2)$,... our total payoff is given by

$$V(\mathbf{h}) = R(\mathbf{h}(0)) + \gamma R(\mathbf{h}(1)) + \gamma^2 R(\mathbf{h}(2)) + \cdots \tag{8}$$

The goal of RL is to choose actions over time so as to maximize the value of payoff. The value function ($V^{\pi}(\mathbf{h})$) defines the expected sum of discounted rewards the controller will receive upon executing a fixed policy π starting from state $\mathbf{h(0)}$ till reaching the desired state h_{desired}.

$$V^{\pi}(\mathbf{h}) = E[R(\mathbf{h(0)}) + \gamma R(\mathbf{h(1)}) + \gamma^2 R(\mathbf{h(2)}) + \cdots | \mathbf{h(0)} = \mathbf{h}, \pi] \tag{9}$$

This relationship can also be represented by a Bellman equation as

$$V^{\pi}(\mathbf{h}) = R(\mathbf{h}) + \gamma V^{\pi}(\mathbf{h}') \tag{10}$$

Here, the first term defines the immediate reward for the controller for being in this starting state. The second term represents the sum of future discounted rewards. This Bellman equation is used for finding the optimal value function for each of the $N_1 \times N_2$ states. The optimal value function, given below, is the value achieved when the optimal policy is followed by the controller.

$$V^*(\mathbf{h}) = \max_{\pi} V^{\pi}(\mathbf{h}) = R(\mathbf{h}) + \max_{q_1 \in Q_1} \gamma V^*(\mathbf{h}') \tag{11}$$