B. Jaganatha Pandian[1] / Mathew M. Noel[1]

Tracking Control of a Continuous Stirred Tank Reactor Using Direct and Tuned Reinforcement Learning Based Controllers

[1] School of Electrical Engineering, VIT University, Vellore 632014, Tamil Nadu, India, E-mail: jaganathapandian@vit.ac.in. http://orcid.org/0000-0001-9491-9226.

Abstract:
The need for linear model, of the nonlinear system, while tuning controllers limits the use of classic controllers. Also, the tuning procedure involves complex computations. This is further complicated when it is necessary to operate the nonlinear system under different operating constraints. Continues Stirred Tank Reactor (CSTR) is one of those non-linear systems which is studied extensively in control and chemical engineering due to its highly non-linear characteristics and its diverse operating range. This paper proposes two different control schemes based on reinforcement learning algorithm to achieve both servo as well as regulatory control. One approach is the direct application of Reinforcement Learning (RL) with ANN approximation and another is tuning of PID controller parameters using reinforcement learning. The main objective of this paper is to handle multiple set point control for the CSTR system using RL. The temperature of the CSTR system is controlled here for multiple setpoint changes. A comparative study is also done between the two proposed algorithm and from the test result, it is seen that direct RL approach with approximation performs better than tuning a PID using RL as oscillations and overshoot are less for direct RL approach. Also, the learning time for the direct RL based controller is lesser than the later.

Keywords: Artificial neural network, CSTR control, PID tuning, Reinforcement learning, tracking control

DOI: 10.1515/cppm-2017-0040

Received: June 10, 2017; **Revised:** September 25, 2017; **Accepted:** October 14, 2017

1 Introduction

A typical non-linear process, which is diversely used in control research and most chemical industrial applications, is the Continuous Stirred Tank Reactor (CSTR). Controlling the reaction temperature and the chemical composition according to the varying product demand in a CSTR remains a major challenge for control experts [1–3]. Advanced control strategies were tried in the past [4–6] for monitoring and control of CSTR parameters. Most of these control design requires a linear model of the process around the operating point. When it is necessary to operate the nonlinear system under different operating zones, the control design also should be adaptive. Such control designs demand piecewise linear modes of the system and also the calculations are complex in nature.

In the recent past, RL has been a widely researched domain [7–9] to handle nonlinear, dynamic control problems. RL based controllers have exhibited better performance than PID controllers on chemical process control applications [10]. In RL, a learning agent learns to take good control actions for a given situation by maximizing a reward function through interactions with the problem environment. This machine learning approach has been successfully tried in various domains, including system identification, sequential decision-making problems and optimal control problems [11–15] under stochastic conditions.

Generally, RL works with finite and discrete samples in the state and action spaces. The challenges faced while applying RL for control system applications, where state and action spaces are continuous, are discussed and solved using efficient computation techniques [16–18]. The nonlinear function approximating potentiality of Artificial Neural Network (ANN) has been coalesced with RL based controllers to handle continuous state control problems with improved stability. ANN was used to approximate the discrete functions involved in the RL based control problems, like, the action (policy) function, value function or the reward function [19–21] while handling nonlinear continuous control problems. The growing need for controller designs for nonlinear and dynamic systems promoted the research in adaptive control approaches. Adaptive PID controllers were tuned

B. Jaganatha Pandian is the corresponding author.

using RL based approaches to address dynamic control problems like wind turbine control, robotic control and engine control [22–25]. Adaptive neural controllers were also designed to demonstrate stability in tracking control problems for nonlinear and multi-input multi-output processes [26–28].

This paper proposes two RL based adaptive control schemes for a nonlinear chemical process control, where, multiple operating conditions are customary. In this work, temperature control problem of a CSTR model is used for conducting experiments. In the first approach, ANN-RL, a direct RL based controller was trained using value iteration algorithm. For tracking control, the desired system state along with the current system state was used as inputs for the RL agent. The obtained optimal policy function was approximated using an ANN to make it continuous. In the second approach, PID-RL, the RL agent was trained to update a PID controller's parameters according to the current and desired state situations. Observed results indicate the direct RL approach with function approximation performs faster and smoother compared to the RL tuned adaptive PID controller approach.

2 Reinforcement learning

In Reinforcement learning (RL), the agent iteratively learns, through interactions with the working environment, to map situations to optimal actions. The solution of this optimal controller learning problem is Markov Decision Process (MDP), which can be expressed by a 5-tuple with:

- **S** represents a set of state variables (discretization is needed to handle the continuous spaces of state)
- **A** represents set of action variables (discretization is needed to handle continuous action)
- $\mathbf{P_{sa}}$ –state transition probability which represents the distribution over which state variables could transit to if an action is taken in a particular state.
- $\boldsymbol{\gamma} \in [0, 1)$ –represents discount factor
- $R : S \times A \rightarrow \mathbb{R}$ -represents reward function

The main objective of RL is to take actions over time so that the expected total payoff value can be maximized. The expected (**E**) total payoff, also known as the value function **V**, when executing a policy $\boldsymbol{\pi}$ on a system with initial state $\boldsymbol{s_0}$, is defined below in eq. (1):

$$\mathrm{V}^{\pi}(\mathrm{s}) = \mathrm{E}\left[\mathrm{R}(\mathrm{s}_0) + \gamma \mathrm{R}(\mathrm{s}_1) + \gamma^2 \mathrm{R}(\mathrm{s}_2) \ldots | \mathrm{s}_0 = \mathrm{s}, \pi\right] \tag{1}$$

This value function $\mathbf{V^{\pi}(s)}$ is the expected cumulative reward which is discounted by the factor $\boldsymbol{\gamma}$. The optimal value function $\mathbf{V^{*}(s)}$ is the value function obtained while executing the optimal policy, which satisfies the well-known Bellman equations [29]:

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

Where, **R(s)** denotes the immediate reward and the second term denote the maximum, for all actions, of the expected cumulative discounted reward.

The policy function $\boldsymbol{\pi}$**: S→A** does the mapping from the current state to the controller action. The optimal policy function $\boldsymbol{\pi^{*}}$**: S→A** is also defined below in eq. (3):

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

This optimal policy function will maximize the total payoff. This optimal policy could be found by two iterative learning algorithms: policy iteration and value iteration. Policy iteration starts with selecting a policy and evaluating it by calculating the value function for improvement. This evaluation and improvement process repeated till optimal policy is achieved. Alternatively, the value iteration algorithm uses a truncated policy evaluation process. This makes the value iteration algorithm converge faster than the policy iteration algorithm for applications with a large set of action values.

3 Continuous Stirred Tank Reactor (CSTR) process

A Continuous Stirred Tank Reactor (CSTR) model is regarded as one of the most challenging unit operations because of its high non-linearity and large-scale operation. The chemical reaction that takes place inside the CSTR is either endothermic or exothermic and to maintain a constant temperature, the heat produced inside the unit must be added or removed. In the CSTR model considered for experiments, the reaction is an irreversible exothermic one. The heat is removed by the coolant flow through the jacket around the system. The produced heat is removed in terms of the difference in temperature between the fluid of the reactor and the coolant in the jacket. The constant temperature maintenance is a very challenging task because of its physical and chemical behavioral complexity.

The nonlinear system dynamics can be explained by two ordinary differential equations (ODE) which are given below in (4).

$$\frac{dT_R}{dt} = \frac{Q_{IN}\,(T_I - T_R)}{V_C} + k_1 C_A e^{-(E/R)/T_R} + k_2 Q_C \left(1 - e^{-\frac{k_2}{Q_C}}\right)(T_c - T_R) \tag{4}$$

$$\frac{dC_A}{dt} = \frac{Q_{IN}\,(C_I - C_A)}{V_C} - k_0 C_A\, e^{-(E/R)/T_R}$$

Temperature and concentration are the two state variables and the coolant flow rate is considered as the manipulated variable in this process. To get a regulatory response the inlet flow rate is treated as disturbance parameter. The nominal process parameters used for modeling are shown in Table 1.

Table 1: CSTR parameters.

Process variables	Operating values
Product flow rate(Q_{IN})	100 l/min
Input product concentration(C_I)	1 mol/l
Input temperature(T_I)	350 K
Coolant temperature (T_C)	350 K
Container volume(V_C)	100 l
Activation energy term(E/R)	10^4 K
Reaction rate constant (k_0)	7.2*10^{10}/min
Plant constant (k_1)	1.44*10^{13} K l/min/mol
Plant constant (k_2)	0.01/l
Plant constant (k_3)	700 l/min

3.1 ANN-RL controller

The conventional RL algorithm works on system's finite state-action spaces. To make this algorithm work for continuous state space, discretization into finite steps is needed for both state and action spaces. The discretization in state-action space heads to controller errors, which is practically equivalent to quantization errors to digital control frameworks. As a remedy, the number of discretization steps can be increased but this proposal fails to measure because of the exponential increase in discretization. As a consequence, the error in discretization leads the control performance to more oscillation and overshoots near the set point. The impacts of discretization on continuous time frameworks look quite similar to the impacts of estimation errors; since the approximate state is only accessible to achieve control in both of the cases.

This discretization error issue is resolved by exploiting the generalization capability of neural networks; to predict the precise control over continuous state domain from the available information on the discretized state. This ANN-RL algorithm is the first approach which is executed for the CSTR system, for both servo and regulatory mechanism. The neural network approach is applied here directly to learn the optimal policy matrix from the measured matrix of optimal policy, for the discretized state.

In Figure 1 the function **F(Sc, Sd)** is optimal policy function,$\tilde{\pi}^*(Sc, Sd)$, as studied above.

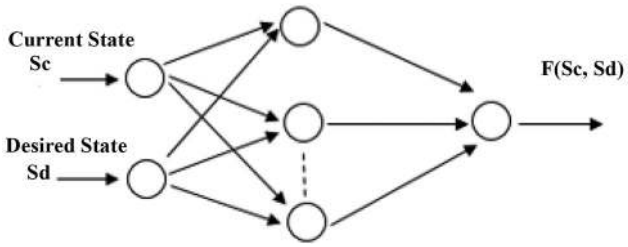

Figure 1: Feed forward neural network structure to learn the best policy.

In this approach, value iteration algorithm is implemented over the discretized states and the training is carried out for the radial basis function (RBF) neural network, with those discretized values. As input and output, the state variables in the continuous domain will be taken by the RBF neural network for control action approximation. The algorithm for this methodology is given in Table 2.

Table 2: ANN-RL algorithm with policy approximation.

1. Define the operating range of T_R, C_A, T_D (desired) and Q_c
2. Initialize R and γ.
3. Discretize T_R, C_A, T_D and Q_c into N_1,N_2, N_3 and N_c levels
4. For each combination of T_R, C_A and T_D initialize V(S):=0
5. Repeat the loop for N number of iterations
   Repeat the loop for all state-action combination
   $V^*(s) = R(s) + \max_{q_c \in Q_c} \gamma V^*(s')$
   End
   End
6. Repeat the loop for all state-action combination
   $\pi^*(s) = \arg\max_{q_c \in Q_c} V^*(s')$
   End
7. Use T_R, C_A, T_D and π^*(s) to train the RBF neural network. Assume $\tilde{\pi}^*(s)$ as the approximation to $\pi^*(s)$ calculated by the RBF neural network.
8. For the continuous control execute the following
   a) Get current state, S_c of the system and desired state S_d
   b) Compute $\tilde{q}_c(s) = \tilde{\pi}^*(s)$
   c) Fix the coolant flow rate at $\tilde{q}_c(s)$
   d) Go to a)

4 RL-PID controller

Despite enormous advancement in control engineering, PID is still considered as the most commonly used controller for many practical cases in control domain because of its simple architecture and robust performance. At every instant, the controller computes the difference between a desired set point and the measured process variable and based on that it will correct the control action.

The PID controller can be represented by the equation given below:

$$u(t) = K_p e(t) + K_i \int_0^t e(\tau)\, d\tau + K_d \frac{de(t)}{dt} \tag{5}$$

To get the best performance for a PID control scheme, the PID controller parameters (K_p, K_i, K_d) have to be determined and adjusted properly. These parameters could be tuned offline or online approaches. In offline tuning, a linearized model of the plant, around the operating point, is used for tuning the controller parameters. But the changing control objective or the change in system dynamics makes it necessary to tune the PID parameter online. There are a number of self-tuning methods proposed for adaptive PID controllers. In this paper, reinforcement learning approach to tune the PID parameters has been suggested.

In the proposed method, the learning agent learns to change the PID parameters for any given system state and for any given desired state. The RL based PID tuning and controller implementation algorithm is given in Table 3.

Table 3: PID tuning using RL.

1. Define the operating range of T_R, C_A, T_D, Kp, Ki and Kd
2. Initialize R and γ.
3. Discretize T_R, C_A, T_D, Kp, Ki and Kd into N_1,N_2, N_3, N_{c1}, N_{c2} and N_{c3} levels
4. For each combination of T_R, C_A and T_D initialize V(S):=0
5. Repeat the loop for N number of iterations
   Repeat the loop for all state-action combination
   $V^*(s) = R(s) + \max_{\{k_p,k_i,k_d\} \in \{Kp, Ki, Kd\}} \gamma V^*(s')$
   End
   End
6. Repeat the loop for all state-action combination
   $\pi^*(s) = \arg\max_{\{k_p,k_i,k_d\} \in \{Kp, Ki, Kd\}} V^*(s')$
   End
7. For the continuous control execute the following
   a) Get current state, S_c of the system and desired state S_d
   b) Compute $\pi^*(s)$
   c) Fix the PID controller parameters at $\pi^*(s)$
   d) Go to a)

5 Results

For testing both control strategies a CSTR system as in (4) was used. The goal is to track the reactor temperature (T_R) as required, by manipulating the flow rate of the coolant around the CSTR jacket. The operating range of the reactor temperature was assumed to be 450 ± 10°K. In the first control learning approach, the learning agent learns to manipulate the coolant flow rate directly based on the current and desired state combination. In the second approach, the agent learns to modify the PID controller parameters {Kp, Ki, Kd} for every current and desired state combination. This PID will manipulate the coolant flow for continuous control.

5.1 Direct RL with ANN (ANN-RL)

In Reinforcement Learning, the MDP starts with the discretization of all continuous variables. State variables (T_R, C_A), the desired variable (T_D) and the action variable (Q_c) were discretized into 20 steps. This gives 8000 combinations of current and desired state variable with 20 possible actions for MDP to work with. So the dimensions of the value function and policy function were 20 × 20 × 20 each. A feed-forward neural network with three inputs (T_R, C_A, T_D) and one output (Q_c) was trained to learn the discontinuous best policy function. During the implementation, the sensors will feed T_R and C_A to the network and the required T_D value fed to the network by the user. The network will give necessary Q_c value for tracking or regulating control.

The system response, for tracking control, obtained with RL controller with policy function approximation is shown below. The desired state and actual state variables are shown in Figure 2. The corresponding action taken is shown in Figure 3.

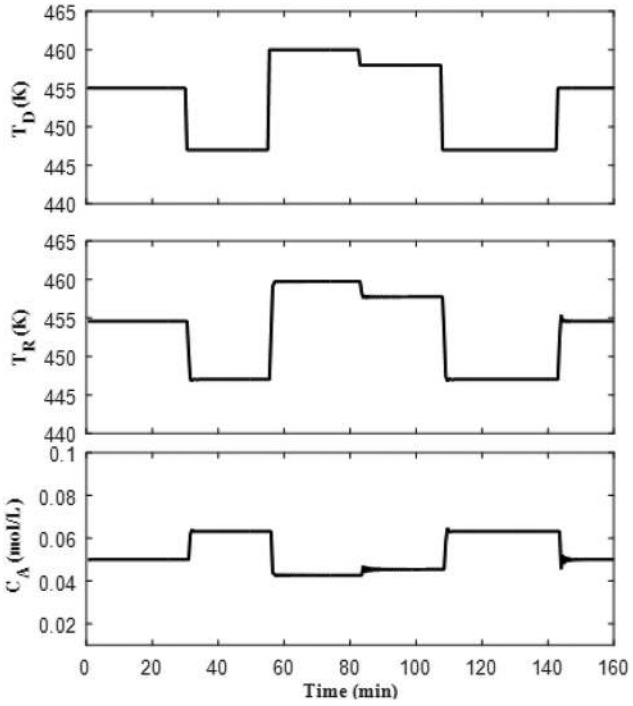

Figure 2: Tracking temperature control response of CSTR under ANN-RL control.

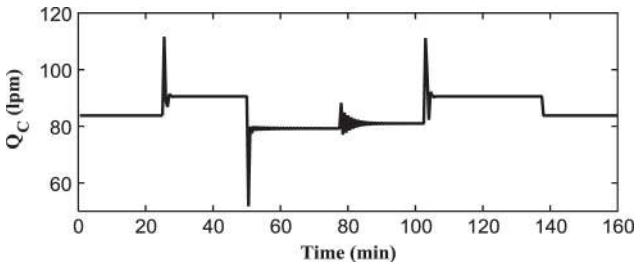

Figure 3: Action taken by ANN-RL during tracking control.

To check the robustness of the above algorithm, regulatory response is also considered in this context. Here, the inlet flow rate (Q_{IN}) is taken as the disturbance parameter and the variation given to it is plotted in Figure 4, along with the system response. From Figure 4, it is clear that the system is able to track the desired temperature (460K) easily without much oscillation after introducing disturbance to the system. Figure 5 shows the action taken by the controller to maintain the desired temperature when the system was disturbed.

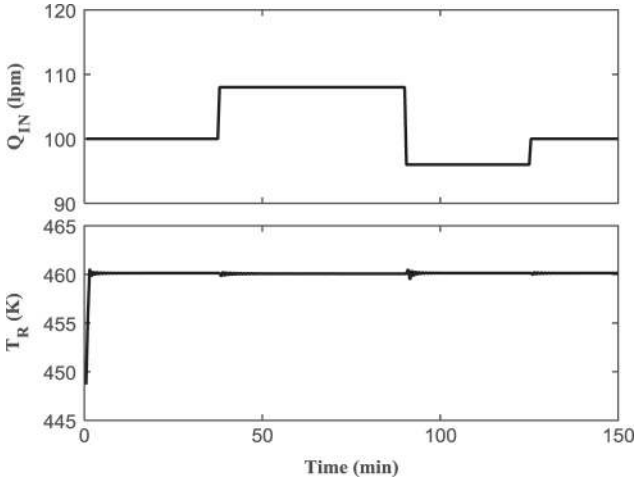

Figure 4: Regulation of CSTR temperature under ANN-RL control.

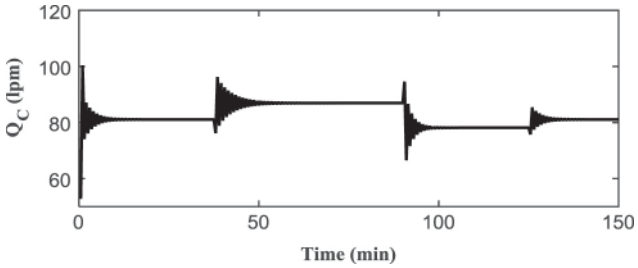

Figure 5: Action taken by ANN-RL during regulatory control.

5.2 RL tuned PID (RL-PID)

The objective here is to train a control agent to adapt the PID controller parameters according to the current and desired state variables. For MDP, the state and desired state variables were discretized as in the previous approach. This gives 8000 possible state variable combinations. The search range for the proportional (***Kp***), integral (***Ki***) and derivative (***Kd***) control gains were fixed between −10 to 10. These are the action variables for the RL agent. So each of them were discretized into 20 steps, which gives 8000 controller settings for the RL agent to choose and implement.

The system response, for tracking control, obtained with RL tuned PID controller is shown below. The desired state and actual state variables are shown in Figure 6. The corresponding action taken by the PID controller is shown in Figure 7.

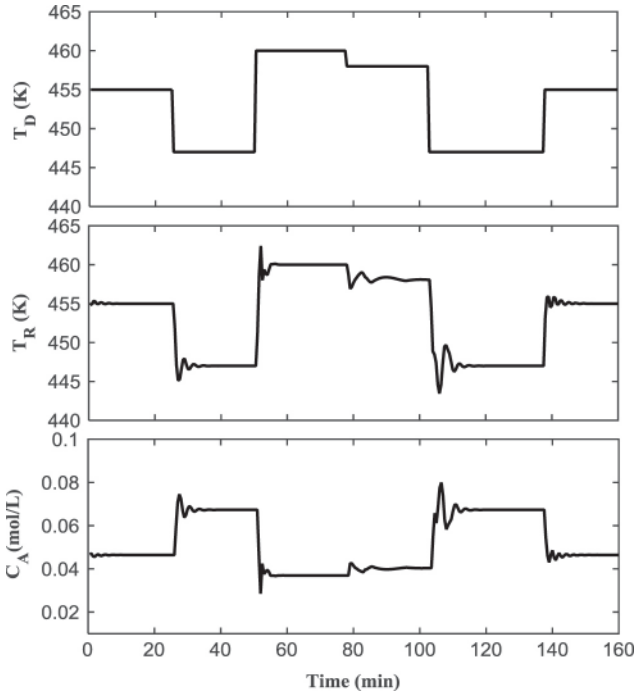

Figure 6: Tracking temperature control response of CSTR under RL-PID control.

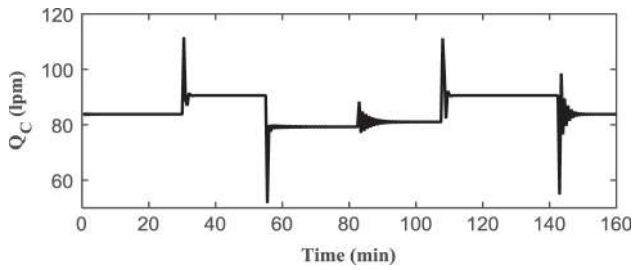

Figure 7: Action taken by RL-PID during tracking control.

To show the action taken by the RL agent in this approach, a portion of the response is taken and shown below. The variation in reactor temperature and the corresponding changes in the PID controller parameters are shown in Figure 8.

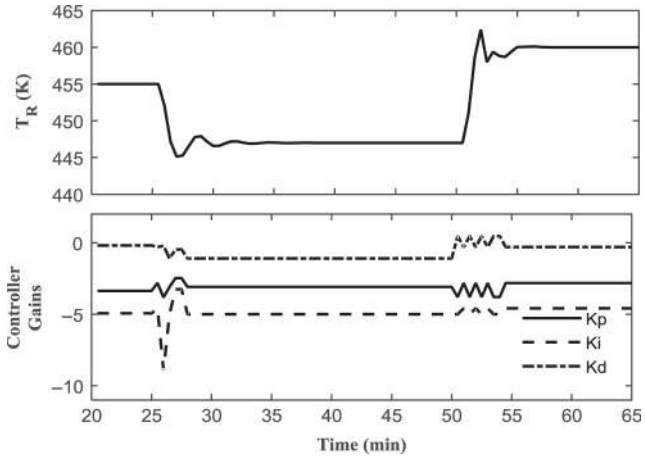

Figure 8: PID parameter variations.

For a regulatory response, a similar disturbance was introduced in the chemical inlet flow rate (Q_{IN}) as in the previous approach. The desired reactor temperature was set at 460°K. The system response under inlet disturbance and the corresponding action taken by the RL tuned PID controller are shown in Figure 9 and Figure 10 respectively.

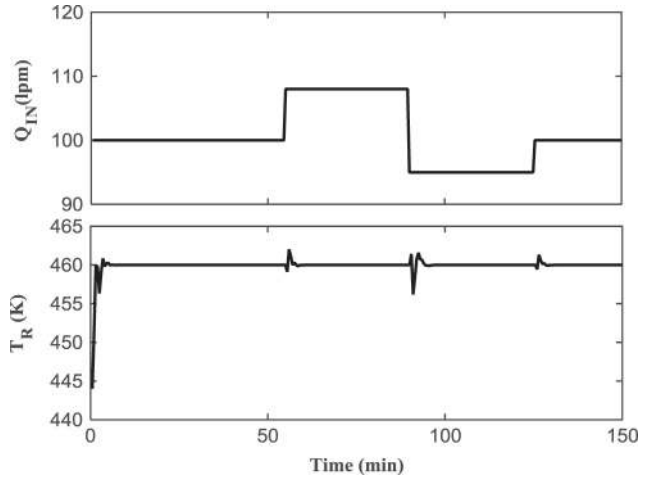

Figure 9: Regulation of CSTR temperature under RL-PID control.

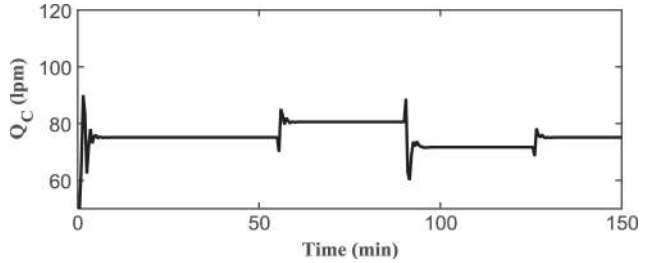

Figure 10: Action taken by RL-PID during regulatory control.

The performances of the controllers were evaluated both in the learning phase and implementation phase. During the learning phase, the time taken to complete one iteration and the time taken for the entire learning to converge were measured. It was observed that ANN-RL approach takes lesser time per iteration and also to converge. In the RL-PID approach, the RL agent takes action on the PID controller parameters which in turn modifies the controller output. This makes RL-PID slower than ANN-RL during the learning phase. In the implementation phase the peak overshoot, settling time and the rate at which the oscillation decay were measured during both servo and regulatory response analysis. The ANN-RL controller ensures lesser overshoot

and faster settling time while the RL-PID controller gives better decaying of process variable oscillations. The worst case observations after multiple testing on the controller performances are given in Table 4.

Table 4: Controller performance assessment.

	Learning phase		Servo response			Regulatory response		
	Time per iteration (min)	Convergence time (min)	Overshoot (%)	Settling time (min)	Decay Ratio	Overshoot (%)	Settling time (min)	Decay Ratio
ANN-RL	0.3	26.208	0.13	4.5	1/2	0.176	4	1/4
RL-PID	2.4	288.0	0.826	5.5	1/3.8	0.783	10	1/5.42

6 Conclusion

Most Reinforcement learning problems are single goal-oriented, where the learning agent takes the current state of the system as input and learns to take optimal action through iterative learning. This paper presents two different RL based approaches for temperature control of nonlinear CSTR process, where multiple goal tracking is necessary. In the first approach, the learning agent takes the current state and the desired state as the input and learns to take direct control actions on the system. To minimize the effects of discretization, which was necessary to handle continuous variable for MDP, an ANN based policy function approximation was used. In the second approach, the RL agent learns to modify a PID controller parameters based on the current and desired state conditions. It is observed that the direct RL with ANN gives smoother transitions for tracking control compared to that of an RL tuned adaptive PID controller. Also, the time taken by the RL tuned PID during the learning and control execution phases are more compared to the direct RL approach. A major challenge for the proposed approaches is the number of state variables in the system under control. Commonly the RL based approaches suffer the curse of dimensionality and the proposed approach adds one more dimension in the form of the "desired state". This might increase the learning time while working with higher order systems. This increase in the problem dimension could be avoided by exploring a time-varying reward function approach for handling similar tasks.

A Nomenclature

a, A Action variable and its constraint set
s, S State vector and its constraint set
R(s) Reward function
P_{sa} Probability of reaching "s" upon execution of "a"
$V^{\pi}(s)$ Cumulative discounted reward
π^{*} Optimal policy
$V^{*}(s)$ Optimal value
Qc Coolant Flow rate (lpm)
C_A Concentration of A in the reactor (mol/l)
T_R Temperature of reactor fluid (K)
Q_{IN} Product Flow rate (lpm)
C_I Input product concentration (mol/lit)
T_I Input temperature (K)
T_C Coolant Temperature (K)
V_C Container volume (l)
E/R Activation energy term (K)
k_0 Reaction rate constant (1pm)
k_1, k_2, k_3 CSTR Plant constants

References

[1] Mohammadzaheri M, Chen L. Intelligent control of a nonlinear tank reactor based on Lyapunov direct method. In: Industrial Technology, 2009. ICIT 2009. IEEE International Conference on 2009 Feb 10:1–6. IEEE.
[2] Salahshoor K, Sabet Kamalabady A. Adaptive feedback linearization control of SISO nonlinear processes using a self-generating neural network-based approach. Chem Prod Process Model. 2011;6(1). DOI: 10.2202/1934-2659.1518
[3] Rahmat MF, Yazdani AM, Movahed MA, Mahmoudzadeh S. Temperature control of a continuous stirred tank reactor by means of two different intelligent strategies. Int J Smart Sens Intell Syst. 2011;4(2):244–67.
[4] Wahab A, Khairi A, Hussain MA, Omar R. An artificial intelligence software-based controller for temperature control of a partially simulated chemical reactor system. Chem Prod Process Model. 2008;3(1):53.
[5] Aguilar R, Poznyak A. A new robust sliding-mode observer design for monitoring in chemical reactors. Analysis. 2004;3:6.
[6] Manimozhi M, Meenakshi R. Multiloop IMC-based PID controller for CSTR process. In: Proceedings of the International Conference on Soft Computing Systems 2016:615–25. Springer India.
[7] Zhang Y, Ding SX, Yang Y, Li L. Data-driven design of two-degree-of-freedom controllers using reinforcement learning techniques. IET Control Theory Appl. 2015;9(7):1011–21.
[8] Radac MB, Precup RE, Roman RC. Model-free control performance improvement using virtual reference feedback tuning and reinforcement Q-learning. Int J Syst Sci. 2017;48(5):1071–83.
[9] Si J, Wang YT. Online learning control by association and reinforcement. IEEE Trans Neural Networks. 2001;12(2):264–76.
[10] Syafiie S, Tadeo F, Martinez E. Model-free learning control of neutralization processes using reinforcement learning. Eng Appl Artif Intell. 2007 Sep 30;20(6):767–82.
[11] Cerrada M, Aguilar J. Reinforcement learning in system identification. California: INTECH Open Access Publisher, 2008.
[12] Govindhasamy JJ, McLoone SF, Irwin GW. Reinforcement learning for process identification, control and optimisation. In: Intelligent Systems, 2004. Proceedings. 2004 2nd International IEEE Conference 2004 Jun 22;1:316–21. IEEE.
[13] Malikopoulos AA, Papalambros PY, Assanis DN. A real-time computational learning model for sequential decision-making problems under uncertainty. J Dyn Syst Meas Control. 2009;131(4):041010.
[14] Wong WC, Lee JH. A reinforcement learning-based scheme for direct adaptive optimal control of linear stochastic systems. Optimal Control Appl Methods. 2010;31(4):365–74.
[15] Pradeep DJ, Noel MM, Arun N. Nonlinear control of a boost converter using a robust regression based reinforcement learning algorithm. Eng Appl Artif Intell. 2016;52:1–9.
[16] Pazis J, Lagoudakis MG. Learning continuous-action control policies. In: Adaptive Dynamic Programming and Reinforcement Learning, 2009. ADPRL'09. IEEE Symposium on 2009 Mar 30:169–76. IEEE.
[17] Weinstein A. Local planning for continuous Markov decision processes. New Jersey: Rutgers The State University of New Jersey-New Brunswick, 2014.
[18] Howell MN, Frost GP, Gordon TJ, Wu QH. Continuous action reinforcement learning applied to vehicle suspension control. Mechatronics. 1997;7(3):263–76.
[19] Lee M, Anderson CW. Convergent reinforcement learning control with neural networks and continuous action search. In: Adaptive Dynamic Programming and Reinforcement Learning (ADPRL), 2014 IEEE Symposium on 2014 Dec 9:1–8. IEEE.
[20] Noel MM, Pandian BJ. Control of a nonlinear liquid level system using a new artificial neural network based reinforcement learning approach. Appl Soft Comput. 2014;23:444–51.
[21] Liu YJ, Tang L, Tong S, Chen CP, Li DJ. Reinforcement learning design-based adaptive tracking control with less learning parameters for nonlinear discrete-time MIMO systems. IEEE Trans Neural Networks Learn Syst. 2015;26(1):165–76.
[22] Howell MN, Best MC. On-line PID tuning for engine idle-speed control using continuous action reinforcement learning automata. Control Eng Pract. 2000;8(2):147–54.
[23] Chamsai T, Jirawattana P, Radpukdee T. Robust adaptive PID controller for a class of uncertain nonlinear systems: an application for speed tracking control of an SI engine. Math Probl Eng. 2015;2015:1–12.
[24] El Hakim A, Hindersah H, Rijanto E. Application of reinforcement learning on self-tuning pid controller for soccer robot multi-agent system. In: Rural Information & Communication Technology and Electric-Vehicle Technology (rICT & ICeV-T), 2013 Joint International Conference on 2013:1–6. IEEE.
[25] Sedighizadeh M, Rezazadeh A. Adaptive PID controller based on reinforcement learning for wind turbine control. Proceedings of World Academy of Science, Engineering and Technology. Cairo, Egypt. 2008;27:257–62.
[26] Liu YJ, Tong S. Optimal control-based adaptive NN design for a class of nonlinear discrete-time block-triangular systems. IEEE Trans Cybern. 2016;46(11):2670–80.
[27] Li DP, Li DJ. Adaptive neural tracking control for nonlinear time-delay systems with full state constraints. IEEE Trans Syst Man Cybern Syst. 2017;47(7):1590–1601.
[28] Li DP, Li DJ, Liu YJ, Tong S, Chen CP. Approximation-based adaptive neural tracking control of nonlinear MIMO unknown time-varying delay systems with full state constraints. IEEE Trans Cybern. 2017;47(10):3100–09.
[29] Bellman RI. Dynamic programming. Princeton, NJ: Princeton University Press,1957:3. 1(2).