Neural $\mathcal{H}_2$ Control Using Reinforcement Learning for Unknown Nonlinear Systems

Adolfo Perrusquía
Departamento de Control Automático
CINVESTAV-IPN
Mexico City, Mexico

Wen Yu
Departamento de Control Automático
CINVESTAV-IPN
Mexico City, Mexico
yuw@ctrl.cinvestav.mx

***Abstract*—In this paper we discuss discrete-time $\mathcal{H}_2$ control for unknown nonlinear system. We use recurrent neural networks to model the system identification, then apply $\mathcal{H}_2$ tracking control. The neural networks based critic control does not require the system dynamics. Our optimal control policy uses a recursive solution of the discrete algebraic Riccati equation and reinforcement learning. The stabilities of system identification and $\mathcal{H}_2$ tracking control are proven. he convergence of the approach is also given by the use of Lyapunov stability theory. The proposed method is validated with the control of a surge tank.**

***Index Terms*—neural control, reinforcement learning**

I. Introduction

Control of discrete-time systems is becoming important in recent years since almost all of the control schemes are implemented on digital devices. Also machine learning techniques are developed in discrete time and are designed according to an optimization problem. Discrete optimal control is a well known control philosophy which its main aim is to find a controller that minimizes/maximizes a certain cost function according to a desired performance [1].

The most popular approaches of optimal controllers are designed in $\mathcal{H}_2$ sense such as the linear quadratic regulator (LQR) control. Here it is used the system dynamics to compute (off-line) the controller that minimizes/maximizes a cost function. To obtain the optimal controller on-line we can use the Hewer algorithm or Lyapunov recursions [2].

When the system dynamics is unknown, the classical $\mathcal{H}_2$ control cannot be applied directly and require other techniques [3]. To obtain an optimal control for unknown system dynamics, adaptive dynamic programming (ADP) or reinforcement learning (RL) is proposed [4]–[6]. This method is model-free or uses partial knowledge of the system dynamics [7]–[9] to obtain on-line the optimal control policy [10]–[12] by using data measured along the system trajectories. Some of its most famous methods are Q-learning, Sarsa, and actor critic.

Q-learning and Sarsa are temporal difference methods of RL [5], [13] that can obtain on-line the optimal control policy using a value or policy iteration method; they are also called critic methods. Actor-critic is a policy search method that uses two separate functions to obtain the optimal value function and policy, respectively. Those RL methods are designed in discrete time and need approximators to deal with large state-action space such as Gaussian kernels, linear parametrizations, neural networks, among others [14]–[16]. Those approximators have good results, nevertheless they need big learning time due the exploration phase of the large input space [17]–[19].

In order to accelerate the learning time, some authors proposed to use a long-short term memory such as eligibility traces [23] to take into account the visited states in previous steps. Other methods use model learning [20], [21] or reference-model learning [22], [23] where the learned model serves as experience and exploit its knowledge for a fast bootstrapping. This kind of methods need accurate approximators to obtain a reliable solution of the optimal control problem. Neural networks is one of the most wide used approximator for RL methods and model-learning approaches. The main advantage of neural networks is that they can estimate and control complex systems by its feedback principle [24], [25]. However they began learning from scratch and need an exploration term as a persistent excitation (PE) signal for a good identification [12], [18], [19].

Another methodology that can accelerate the learning time and that is not well established at the current literature is the use of recurrent neural networks (RNN) [26]–[28] as a model learning method. Here the user proposes a stable dynamics at the identification system such that it serves for the weights update. Usually for discrete time systems, it is used the gradient descent rule [29], [30] and robust modifications [31] to guarantee the system identification with small bounded error, however for control purposes, the neural identificator is sensitive to modelling error and cannot guarantee optimal performance.

In this paper we proposed a critic learning based on recurrent neural networks for discrete-time nonlinear system identification and tracking control. The proposed method is designed to obtain a solution of the $\mathcal{H}_2$ control problem. The neural identificator is based on a parallel recurrent neural network which provides an easy way to compute the optimal solution recursively. In order to avoid sensitivity of the controller against modellling error, it is proposed to use a reinforcement learning based on a neural network approximator. Stability and convergence of the proposed method is proven via Lyapunov stability theory. Simulations studies are carried out in the control of a surge tank. The results show optimal solutions with bounded error without knowledge of the nonlinear system dynamics.

978-1-7281-6926-2/20/$31.00 ©2020 IEEE

II. Unknown Nonlinear System Modeling

Consider the following-discrete time state-space non-linear system

$$x_{k+1} = f(x_k, u_k), \tag{1}$$

where $u_k \in \mathbb{R}^m$ is the control input, $x_k \in \mathbb{R}^n$ is the state vector, and $f : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^n$ is the transition function or dynamics.

Consider the discrete-time serial-parallel recurrent neural network [30]:

$$\widehat{x}_{k+1} = A\widehat{x}_k + W_{1,k}\sigma(W_{2,k}x_k) + U_k \tag{2}$$

where $\widehat{x}_k \in \mathbb{R}^n$ is the state of the RNN. The matriz $A \in \mathbb{R}^{n\times n}$ is a stable Hurwitz matrix which will be specified after. The matrices $W_{1,k} \in \mathbb{R}^{n\times r}$ is the weight matrix of the output weights and $W_{2,k} \in \mathbb{R}^{l\times n}$ is the weights matrix of the hidden layer. $U_k = [u_1, u_2, \cdots, u_m, 0, \cdots, 0]^\top \in \mathbb{R}^n$ is the control action. $\sigma(\cdot) : \mathbb{R}^l \to \mathbb{R}^r$ is the activation function. The elements of $\sigma_i(\cdot)$ can be any stationary, bounded and monotone increasing functions.

Using the RNN identificator, the critic controller is designed to guarantee the tracking of a desired reference $x_d \in \mathbb{R}^n$. The desired trajectory is regarded as the solution of the following discrete model:

$$x_{d,k+1} = \varphi(x_{d.k}) \tag{3}$$

In the following sections the on-line critic learning is developed using tow approaches: recurrent neural networks (RNN) solution and reinforcement learning (RL) solution. Both approaches are designed for an exact model matching case and modelling error case.

A. Exact model match

Consider that the RNN can exactly approximate the non-linear system (1). According to the Stone-Weierstrass theorem [32], the non-linear model (1) can be written as follows:

$$x_{k+1} = Ax_k + W_1^*\sigma(W_2^*x_k) + U_k \tag{4}$$

where W_1^* and W_2^* are optimal weight matrices. In order to obtain the parameters update rule we expand the second term of the right side of (2) using the Taylor formula [29]:

$$g(x) = \sum_{i=0}^{l-1} \frac{1}{i!}\left[(x_1 - x_1^0)\frac{\partial}{\partial x_1} + (x_2 - x_2^0)\frac{\partial}{\partial x_2}\right]_0^i g(x^0) + \varepsilon_g,$$

where $x = [x_1, x_2]^\top$ and $x^0 = [x_1^0, x_2^0]^\top$, ε_g is the remainder of the Taylor formula. Let x_1 and x_2 correspond to $W_{1,k}$ and $W_{2,k}x_k$, x_1^0 and x_2^0 correspond to W_1^* and $W_2^*x_k$, respectively, and $l = 2$. The identification error is defined as:

$$\widetilde{x}_k = \widehat{x}_k - x_k.$$

Then from (4) and (2)

$$\widetilde{x}_{k+1} = A\widetilde{x}_k + \widetilde{W}_{1,k}\sigma(W_{2,k}x_k) + W_1\sigma'\widetilde{W}_{2,k}x_k + \varepsilon_W \tag{5}$$

where $\widetilde{W}_{1,k} = W_{1,k} - W_1^*$, $\widetilde{W}_{2,k} = W_{2,k} - W_2^*$ and ε_W is a second order approximation error of the Taylor series.

Assumption 1: There exists a constant $\beta \geq 1$ such that:

$$\|x_{k+1}\| \geq \frac{1}{\beta}\|x_k\|. \tag{6}$$

Remark 1: The condition $\|x_{k+1}\| \geq \frac{1}{\beta}\|x_k\|$ is a dead zone [31]. If β is selected big enough, the dead-zone becomes small. The dead-zone helps the neural network to update the weights only in small zones where (6) is satisfied. Otherwise, the weights are updated only in the cases where $\|x_{k+1}\| \geq \|x_k\|$ is satisfied.

Assumption 2: If the eigenvalues of the matrix A are between the interval $-\frac{1}{\beta} < \lambda(A) < 0$, then for any matrix $Q = Q^\top > 0$ there exists an unique solution $P = P^\top > 0$ to the following Lyapunov equation:

$$A^\top PA - P + Q + \left(\frac{1}{\beta}I + A\right)^\top P\left(\frac{1}{\beta}I + A\right) = 0. \tag{7}$$

The above Lyapunov equation can be solved on-line as:

$$\left[I - A^\top \otimes A^\top - \left[\tfrac{1}{\beta}I + A\right]^\top \otimes \left[\tfrac{1}{\beta}I + A\right]^\top\right] \text{vec}(P) = \text{vec}(Q)$$

where $\otimes$ is the Kronecker product, and $\text{vec}(\cdot)$ is the matrix stretch.

B. With unmodeled dynamic

Generally, the neural network cannot fully describe the nonlinear system (1). The modeling error is defined as

$$\varepsilon_f = f(x_k, u_k) - A\widehat{x}_k - W_1^*\sigma(W_2^*\widehat{x}_k). \tag{8}$$

The dynamics of the nonlinear system (1) can be written as

$$x_{k+1} = Ax_k + W_{1,k}\sigma(W_{2,k}\widehat{x}_k) + U_k + \xi_k \tag{9}$$

where $\xi_k = \varepsilon_f + W_{1,k}[\sigma(W_{2,k}x_k) - \sigma(W_{2,k}\widehat{x}_k)]$ is also bounded. The dynamics of the identification error (10) becomes

$$\widetilde{x}_{k+1} = A\widetilde{x}_k + \widetilde{W}_{1,k}\sigma(W_{2,k}\widehat{x}_k) + W_{1,k}\sigma'\widetilde{W}_{2,k}x_k + \xi_k$$

III. Neural $\mathcal{H}_2$ Control Using Reinforcement Learning

Now consider the tracking control problem using the RNN. The non-linear dynamics is expressed as:

$$x_{k+1} = Ax_k + W_{1,k}\sigma(W_{2,k}x_k) + U_k + \zeta_k \tag{10}$$

where $\zeta_k = \widetilde{W}_{1,k}\sigma(W_{2,k}x_k) + W_{1,k}\sigma'\widetilde{W}_{2,k}x_k + \varepsilon_k$. Let define the tracking error as

$$e_k = x_k - x_{d,k}.$$

If the control input U_k is chosen as

$$U_k = U_1 + U_2 \tag{11}$$

$$= \varphi(x_{d,k}) - Ax_{d,k} - W_{1,k}\sigma(W_{2,k}x_k) + U_2, \tag{12}$$

where U_2 is a state-feedback controller. Then the tracking error dynamics is simplified to:

$$e_{k+1} = Ae_k + U_2 + \zeta_k. \tag{13}$$

The state-feedback controller U_2 has the form of [28]

$$U_2 = -Ke_k = -(R_c + P_c)^{-1} P_c A e_k, \tag{14}$$

where $R_c = R_c^\top > 0$, which satisfies the following assumption:

Assumption 3: There exists a strictly positive definite matrix Q_c such that the discrete algebraic Riccati equation (DARE)

$$A^\top P_c A + Q_c - A^\top P_c (R_c + P_c)^{-1} P_c A - P_c = 0 \tag{15}$$

has a positive solution $P_c = P_c^\top > 0$.

The solution of the above Riccati equation can be obtained iteratively using Lyapunov recursions [2] as:

$$P_c^{j+1} = (A - K)^\top P_c^j (A - K) + Q_c + K^\top R K,$$

where j is the step index. Since $(A - K)$ is stable, then the recursion converges to the solution of the Riccati equation for any choice of initial value P_c^0.

Assumptions 2 and 3 define the Recurrent Neural Networks solution for the critic learning control. The following theorem gives the learning procedure and the convergence of the critic learning to the desired reference.

Theorem 1: Consider the non-linear dynamics (1), the reference (3) and the model matching neural network (2), whose weights are adjusted by

$$\begin{aligned} \widetilde{W}_{1,k+1} &= \widetilde{W}_{1,k} - \eta \left(P A \widetilde{x}_k + W_1 \sigma' \widetilde{W}_{2,k} x_k \right) \sigma^\top \\ \widetilde{W}_{2,k+1} &= \widetilde{W}_{2,k} - \eta \sigma'^\top W_{1,k}^\top P A \widetilde{x}_k x_k^\top \end{aligned} \tag{16}$$

where η satisfies:

$$\eta = \begin{cases} \frac{\eta_0}{1 + \|PA\sigma\|^2 + \|\sigma'^\top W_{1,k}^\top P A x_k\|^2} & \text{if } \|\widetilde{x}_{k+1}\| \geq \frac{1}{\beta} \|\widetilde{x}_k\| \\ 0 & \text{other case,} \end{cases} \tag{17}$$

with $\eta_0 \in (0, 1]$; and assume that assumptions 1, 2 and 3 are satisfied. Then the identification error and the tracking error converge globally asymptotically to zero.

Proof 1: Consider the Lyapunov function:

$$V_k = \widetilde{x}_k^\top P \widetilde{x}_k + e_k^\top P_c e_k + \frac{1}{\eta} \left(\text{tr}(\widetilde{W}_{1,k}^\top \widetilde{W}_{1,k}) + \text{tr}(\widetilde{W}_{2,k}^\top \widetilde{W}_{2,k}) \right) \tag{18}$$

The time difference of the Lyapunov equation, $\Delta V_k = V_{k+1} - V_k$ is:

$$\begin{aligned} \Delta V_k =& \widetilde{x}_{k+1}^\top P \widetilde{x}_{k+1} + e_{k+1}^\top P_c e_{k+1} - \widetilde{x}_k^\top P \widetilde{x}_k - e_k^\top P_c e_k \\ &+ \frac{1}{\eta} \left(\text{tr}(\widetilde{W}_{1,k+1}^\top \widetilde{W}_{1,k+1}) + \text{tr}(\widetilde{W}_{2,k+1}^\top \widetilde{W}_{2,k+1}) \right) \\ &- \frac{1}{\eta} \left(\text{tr}(\widetilde{W}_{1,k}^\top \widetilde{W}_{1,k}) + \text{tr}(\widetilde{W}_{2,k}^\top \widetilde{W}_{2,k}) \right) \end{aligned}$$

Consider only the identification components $\Delta V_{1,k} = \Delta V_k - e_{k+1}^\top P_c e_{k+1} + e_k^\top P_c e_k$. Substituting the identification error dynamics (5) on $\Delta V_{1,k}$ yields:

$$\begin{aligned} \Delta V_{1,k} =& \widetilde{x}_k^\top (A^\top P A - P) \widetilde{x}_k + \sigma^\top \widetilde{W}_{1,k}^\top P \widetilde{W}_{1,k} \sigma \\ &+ 2 \widetilde{x}^\top A^\top P (\widetilde{W}_{1,k} \sigma + W_{1,k} \sigma' \widetilde{W}_{2,t} x_k + \varepsilon_W) \\ &+ 2 \sigma^\top \widetilde{W}_{1,k}^\top P (W_{1,k} \sigma' \widetilde{W}_2 x_k + \varepsilon_W) + \varepsilon_W^\top P \varepsilon_W \\ &+ x_k^\top \widetilde{W}_{2,k}^\top \sigma'^\top W_{1,k}^\top P W_{1,k} \sigma' \widetilde{W}_{2,k} x_k + \eta \text{tr}(Z_1^\top Z_1) \\ &+ 2 \widetilde{x}^\top \widetilde{W}_{2,k}^\top \sigma'^\top W_{1,k}^\top P \varepsilon_W - 2 \text{tr}(\widetilde{W}_{1,k}^\top Z_1) \\ &- 2 \text{tr}(\widetilde{W}_{2,k}^\top Z_2) + \eta \text{tr}(Z_2^\top Z_2) \end{aligned}$$

where $Z_1 = (PA\widetilde{x}_k + PW_1 \sigma' \widetilde{W}_{2,k} x_k) \sigma^\top$ and $Z_2 = \sigma'^\top W_{1,k}^\top P A \widetilde{x}_k x_k^\top$. The above expression is simplified by using the assumption 2 and the Minkowski inequality as

$$\begin{aligned} \Delta V_{1,k} \leq& \widetilde{x}_k^\top (A^\top P A - P) \widetilde{x}_k + \|P\| \|\widetilde{x}_{k+1} - A \widetilde{x}_k\|^2 \\ &+ 2 \varepsilon_W^\top P (\widetilde{x}_{k+1} - \varepsilon_W) + \eta \|\sigma'^\top W_{1,k}^\top P A \widetilde{x}_k x_k^\top\|^2 \\ &+ \eta \|(PA\widetilde{x}_k + PW_{1,k} \sigma' \widetilde{W}_{2,k} x_k) \sigma^\top\|^2 \end{aligned}$$

The second term of the above expression can be written as:

$$\begin{aligned} \|P\| \|\widetilde{x}_{k+1} - A \widetilde{x}_k\|^2 &\leq \|P\| (\|\widetilde{x}_{k+1}\|^2 + \|A \widetilde{x}_k\|^2) \\ &\leq \|P\| \left(\frac{1}{\beta^2} + \|A\|^2 \right) \|\widetilde{x}_k\|^2 \\ &\leq \|P\| \left\| \frac{1}{\beta} I + A \right\|^2 \|\widetilde{x}_k\|^2 \\ &= \widetilde{x}_k^\top \left(\frac{1}{\beta} I + A \right)^\top P \left(\frac{1}{\beta} I + A \right) \widetilde{x}_k \end{aligned}$$

Note that the second term of the output weights update rule asymptotically converges to zero since it depends on the error of the hidden layer weights. Then

$$\begin{aligned} \Delta V_{1,k} \leq& - \left(\lambda_m(Q) - \frac{1}{\beta^2} \right. \\ &\left. - \eta_0 \frac{\|PA\sigma\|^2 + \|\sigma'^\top W_{1,k}^\top P A x_k\|^2}{1 + \|PA\sigma\|^2 + \|\sigma'^\top W_{1,k}^\top P A x_k\|^2} \right) \|\widetilde{x}_k\|^2 \\ &+ (\lambda_M^2(P) - 2\lambda_m(P)) \|\varepsilon_W\|^2. \\ \Delta V_{1,k} \leq& - \Xi \|\widetilde{x}_k\|^2 + \Pi \|\varepsilon_W\|^2. \end{aligned}$$

where

$$\begin{aligned} \Xi &= \lambda_m(Q) - \frac{1}{\beta^2} - \frac{\eta_0 \kappa}{1 + \kappa}, \\ \kappa &= \max_k (\|PA\sigma\|^2 + \|\sigma'^\top W_{1,k}^\top P A x_k\|^2) \\ \Pi &= \lambda_M^2(P) - 2\lambda_m(P). \end{aligned}$$

where Ξ is positive if $\lambda_m(Q) > \frac{1}{\beta^2} + \frac{\eta_0 \kappa}{1+\kappa}$. Since the second order approximation error depends on powers of the weight errors, then they converge asymptotically to zero as the identification error approximates to zero.

Now consider the control components of ΔV_k,

$$\begin{aligned}\Delta V_{2,k} =& e_{k+1}^\top P_c e_{k+1} - e_k^\top P_c e_k \pm e_k^\top Q_c e_k \pm U_2^\top R_c U_2 \\ =& e_k^\top (A^\top P_c A - P_c \pm Q_c) e_k \pm U_2^\top (R_c + P_c) U_2 \\ &+ 2e_k^\top A^\top P_c (U_2 + \zeta_k) + 2U_2^\top P_c \zeta_k + \zeta_k^\top P_c \zeta_k \\ =& - e_k^\top (Q_c + (R_c + P_c)^{-1}) e_k + \zeta_k^\top P_c \zeta_k \\ &- 2e_k^\top A^\top P_c (R_c + P_c)^{-1} P_c \zeta_k \\ \leq& - e_k^\top (Q_x - A^\top P_c (R_c + P_c)^{-2} P_c A) e_k \\ &+ \zeta_k^\top (P_c + P_c^2) \zeta_k \\ \leq& - e_k^\top \Upsilon e_k + \zeta_k^\top \Omega \zeta_k\end{aligned}$$

where $Q_x = Q_c + (R_c + P_c)^{-1}$, $\Upsilon = Q_x - A^\top P_c (R_c + P_c)^{-2} P_c A$ and $\Omega = P_c + P_c^2$. Q_x is positive definite since $-\frac{1}{\beta} < \lambda(A) < 0$. The perturbation ζ_k is equal to zero since it depends on the NN weights error and the second order approximation error. Hence the tracking error converges asymptotically to zero. This completes the proof.

With unmodeled dynamic, the tracking error dynamics under the controller U_k of (12) is

$$e_{k+1} = Ae_k + U_2 + d_k$$

where $d_k = \widetilde{W}_{1,k}\sigma(W_{2,k}\widehat{x}_k) + W_{1,k}\sigma'\widetilde{W}_{2,k}x_k + \xi_k$.

Theorem 2: Consider the nonlinear system (1), the reference (3) and the recurrent neural network (2), whose weights are adjusted by (16); and assume that assumptions 1-3 are satisfied. Then the identification error and tracking error converge into a small bounded set which implies input-to-state stability (ISS) and semi-global convergence.

Proof 2: Consider the same Lyapunov equation (18). It is used the same procedure of Theorem 1, obtaining:

$$\begin{aligned}&\Delta V_{1,k} \leq -\Xi\|\widetilde{x}_k\|^2 + \Pi\|\xi_k\|^2 \\ &\Delta V_{2,k} - \lambda_m(\Upsilon)\|e_k\|^2 + \lambda_M(\Omega)\|d_k\|^2\end{aligned}$$

The above inequalities satisfy the ISS condition and there exists a big enough Ξ and Υ such that the identification error converges into a small bounded set $\mu_1 = \sqrt{\frac{\Pi}{\Xi}}\|\xi_k\|$ and the tracking error converges into a small bounded set $\mu_2 = \sqrt{\frac{\lambda_M(\Omega)}{\lambda_m(\Upsilon)}}\|d_k\|$.

However the control input U_k of the recurrent neural network is sensitive to modeling error and disturbances. The feedforward term U_1 assumes that it compensates the nonlinear dynamics with some modeling error, but it is well known that a bad design of this controller affects the tracking performance. The optimal control U_2 is a simple linear quadratic controller (LQR) that does not have any information of the real system since it is assumed that the feedforward control term compensates the system nonlinear dynamics. To overcome this issue we use reinforcement learning. Let define the discounted Lyapunov value function [5] as

$$\begin{aligned}V_{2,k} &= \sum_{i=k}^{\infty} \gamma^{i-k} \left(e_i^\top Q_c e_i + U_{2,i}^\top R_c U_{2,i}\right) \\ V_{2,k} &= e_k^\top Q_c e_k + U_2^\top R_c U_2 + \gamma V_{2,k+1}\end{aligned} \tag{19}$$

where $\gamma < 1$ is a discounted factor that guarantees the convergence of the Lyapunov value function. Consider the following NN approximator

$$\hat{V}_{2,k} = \phi^\top(e_k)\theta_k = \phi_k^\top \theta_k \tag{20}$$

where $\theta_l \in \mathbb{R}^p$ is a weight vector and $\phi(\cdot) : \mathbb{R}^n \to \mathbb{R}^p$ are the NN activation functions with p neurons at the hidden layer. The value function $V_{2,k}$ can be rewritten as

$$V_{2,k} = \phi_k^\top \theta^* + \varepsilon(e_k)$$

where θ^* is an optimal weight value and $\varepsilon_k = \varepsilon(e_k)$ is the NN approximation error. Substituting the above expression in (19) yields:

$$\begin{aligned}\phi_k^\top \theta^* + \varepsilon_k &= e_k^\top Q_c e_k + U_2^\top R_c U_2 + \gamma(\phi_{k+1}^\top \theta^* + \varepsilon_{k+1}) \\ \varepsilon(e_k) - \gamma\varepsilon(e_{k+1}) &= e_k^\top Q_c e_k + U_2^\top R_c U_2 + (\gamma\phi_{k+1}^\top - \phi_k^\top)\theta^* \\ H(e_k, U_2, \theta^*) &= \nu_k = r_{k+1} + (\phi_{k+1}^\top - \phi_k^\top)\theta^*\end{aligned} \tag{21}$$

where $\nu_k = \varepsilon_k - \gamma\varepsilon_{k+1}$ is the residual error of the NN approximator which is equivalent to the discrete-time Hamiltonian, and $r_{k+1} = e_k^\top Q_c e_k + U_2^\top R_c U_2$ is the immediate reward or utility function. We want to design an optimal controller which minimizes the residual error using reinforcement learning. Consider the approximate Hamiltonian as:

$$\widehat{H}(e_k, U_2; \theta_k) = \delta_k = r_{k+1} + (\gamma\phi_{k+1}^\top - \phi_k^\top)\theta_k.$$

Here δ_k stands to the temporal difference error of reinforcement learning algorithms, which also can be written as:

$$\delta_k = (\gamma\phi_{k+1}^\top - \phi_k^\top)\widetilde{\theta}_k + \nu_t \tag{22}$$

where $\widetilde{\theta}_k = \theta_k - \theta^*$. Consider the objective function defined as the squared temporal difference error:

$$E = \frac{1}{2}\delta_k^2.$$

We use the normalized gradient descent algorithm [18] for the NN weights update as:

$$\theta_{k+1} = \theta_k - \alpha\frac{\partial E}{\partial \theta_k} = \theta_k - \alpha\delta_k\frac{q_k}{(q_k^\top q_k + 1)^2}, \tag{23}$$

where α is the learning rate and $q_k^\top = (\gamma\phi_{k+1}^\top - \phi_k^\top)$. Here it is used the normalized gradient descent to assure that the weights update are bounded. Also the update rule can be rewritten as:

$$\widetilde{\theta}_{k+1} = \widetilde{\theta}_k - \alpha\frac{q_k q_k^\top}{(q_k^\top q_k + 1)^2}\widetilde{\theta}_k - \alpha\frac{q_k}{(q_k^\top q_k + 1)^2}\nu_k. \tag{24}$$

Assuming that the optimal control is designed as:

$$U_2 = -\frac{1}{2}R_c^{-1}\nabla\phi^\top(e_{k+1})\theta_k, \tag{25}$$

where $\nabla = \partial/\partial e_{t+1}$. Substituting the optimal control (25) at the Hamiltonian yields:

$$\delta_k = e_k^\top Q_c e_k - \frac{1}{4}\theta_k^\top \nabla\phi_{k+1} R_c^{-1} \nabla\phi_{k+1}^\top \theta_k + (\phi_{k+1}^\top - \phi_k^\top)\theta_k.$$

To guarantee convergence of the NN parameters, $\theta_k \to \theta^*$, let introduce the following persistent exciting (PE) definition of discrete-time systems.

Definition 1: Let $q/(q^\top q+1)$ be persistently exciting (PE) in T steps if there exist constants $\beta_1, \beta_2 > 0$, such that

$$\beta_1 I \leq S_1 = \sum_{j=k+1}^{k+T} \frac{q_j q_j^\top}{(q_j^\top q_j + 1)^2} \leq \beta_2 I \tag{26}$$

Lemma 1: Consider the parameters error dynamics (24) be rewritten as a linear time variant (LTV) discrete-system of the form:

$$\begin{aligned} \widetilde{\theta}_{k+1} &= \alpha \frac{q_k}{q_k^\top q_k + 1} u_k \\ y_k &= \frac{q_k^\top}{q_k^\top q + 1} \widetilde{\theta}_k, \end{aligned} \tag{27}$$

where $u_k = -y_k - \frac{q_k}{q_k^\top q_k + 1}\nu_k$ is an output feedback controller. Consider $\frac{q_k}{q_k^\top q_k + 1}$ be PE. Then the parameters error $\|\widetilde{\theta}_k\|$ converges into a bounded set

$$\|\theta_k\| \leq \frac{\sqrt{\beta_2 T}}{\beta_1} \left(\|y_k\| + \alpha\beta_2 (\|y_k\| + \|\nu_k\|) \right) \tag{28}$$

Proof 3: It is similar with the proofs in [29].

Theorem 3: Let $q/(q^\top q+1)$ be persistently exciting (PE). Then there exists contraction mappings $\mathcal{H}$ and $\mathcal{H}'$ with contraction factor γ and γ', respectively, such that the parameters error $\widetilde{\theta}$ and the NN approximator $F(\theta)$ are bounded as:

$$\|\theta - \theta^*\| \leq \frac{1+\gamma}{1-\gamma'} \bar{\nu} \tag{29}$$

$$\|F(\theta) - F(\theta^*)\| \leq \frac{\gamma'(1+\gamma)}{\gamma(1-\gamma')} \bar{\nu}. \tag{30}$$

where $\bar{\nu}$ is an upper bound of the residual error ν_k.

Proof 4: It is similar with the proofs in [29].

Bound (29) is the strictest upper bound that our approach can possess by assuming a rich exploration of the PE signal. However this exploration is limited to a series of limited steps which are not seen at bound (29). In order to see how the PE signal affects the parameters error upper bound, let introduce the following theorem.

Theorem 4: Let U_2 be any admissible control. Let the critic parameters are updated by (24) and assume that $q_k/(q_k^\top q_k + 1)$ is PE. Then the parameters error converge into the following bounded residual set:

$$\|\widetilde{\theta}_k\| \leq \frac{\sqrt{\beta_2 T}\left[(1+\gamma)\gamma' + \alpha\beta_2(\gamma+\gamma')\right]}{\beta_1 \gamma (1-\gamma')} \bar{\nu} \tag{31}$$

Now we are in position to prove Theorem 4.

Proof 5: From Lemma 1 we have the bound (28). The system output y_k is defined by the normalization of the NN approximation $F(\theta)$, then from Theorem 3 we have that

$$\|y_k\| = \left\| \frac{1}{q_k^\top q_k + 1} \left(F(\theta) - F(\theta^*) \right) \right\| \leq \|F(\theta) - F(\theta^*)\|$$

since $q_k^\top q_k + 1 \geq 1$. Then the system output is bounded by (30) as

$$\|y_k\| \leq \frac{\gamma'(1+\gamma)}{\gamma(1-\gamma')} \bar{\nu},$$

and the residual error $\|\nu_k\| \leq \bar{\nu}$. Substituting the output and residual error upper bounds in (28) yields:

$$\widetilde{\theta}_k \leq \frac{\sqrt{\beta_2 T}\left[(1+\gamma)\gamma' + \alpha\beta_2(\gamma+\gamma')\right]}{\beta_1 \gamma (1-\gamma')} \bar{\nu}$$

If the number of neurons at the hidden layer are increased, i.e., $p \to \infty$, then the residual error is decreased $\nu_t \to 0$. Nevertheless, this can cause the overfitting problem in the training of the neural network approximator.

Remark 2: Both the LQR and RL methods use the RNN as a model reference for the control design. The LQR controller is obtained according to the proposed matrix A and assumes that the modelling error and disturbances are small enough such that the control gain compensates them, hence is sensitive to the modelling error. On the other hand, the RL controller learns the control law by considering the complete closed-loop dynamics which includes the modelling error and it only uses the RNN model as previous knowledge. Therefore the RL control law is more robust in comparison to the LQR control law [35].

IV. Surge Tank Example

Consider the surge tank model [33] that is represented by the following differential equation:

$$\frac{dh(t)}{dt} = -\frac{c\sqrt{2gh(t)}}{A_r(h(t))} + \frac{1}{A_r(h(t))} u(t) \tag{32}$$

where $u(t)$ is the input flow, $h(t)$ is the liquid level; $A_r(h(t))$ is the cross-sectional area of the tank; $g = 9.81$ m/s^2 is the gravitational acceleration; $c = 1$ is the known cross-sectional area of the output pipe. Let $A_r(h(t)) = \sqrt{ah(t) + b}$, where $a = 1$ and $b = 3$. Using Euler approximation to discretize the system yields

$$h_{k+1} = h_k + T\left[\frac{-\sqrt{19.62 h_k}}{A_r(h_k)} + \frac{1}{A_r(h_k)} u_k \right] \tag{33}$$

where $T = 0.01$ is the sample time. Here we compare the performance of the RNN solution with our RL solution. The simulation lasts 100 seconds of simulation time.

We choose a scalar $A = -0.5$ and a constant $\beta = 4$ for the recurrent neural network identifier. For the discrete Lyapunov function it is proposed a scalar $Q = 5$ which gives a kernel solution of $P = 7.2727$. For the LQR control design it is proposed the following scalar values: $Q_c = 1$ and $R_c = 0.1$ and a initial value of $P_c^0 = 0.5$. The Lyapunov recursion converges to the DARE solution which is $P_c = 1.023$. It is used 10 hidden nodes at the hidden layer and one node at the output layer. The elements of $W_{1,0} \in \mathbb{R}^{1\times 10}$ and $W_{2,0} \in \mathbb{R}^{10}$ are random numbers between $[0, 1]$. The initial liquid level is $h_0 = 0.2$ and the RNN initial condition is $x_0 = 0$. We use as activation function $\sigma(\cdot) = \tanh(\cdot)$ and therefore $\sigma'(\cdot) = \text{sech}^2(\cdot)$.

The PE signal is chosen as a sum of sine functions which are given at the reference signal as:

$$x_{d,k} = 1.8 + \sin(0.5k) + 0.15\cos(0.3k) + 0.5\sin(0.75k).$$

For the RL solution it is used only one neuron with quadratic activation function, i.e., $\phi_k = e_k^2$ with a learning rate of $\alpha = 0.5$ and a discount factor of $\gamma = 0.9$.

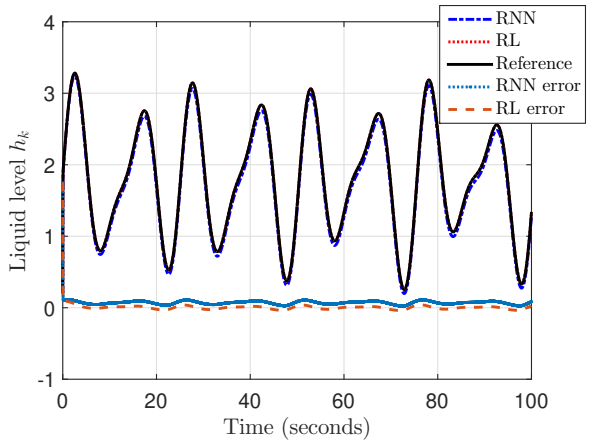


Fig. 1: Tracking control for the surge tank

The tracking results are given in Fig. 1 and the control input in Fig. 2. The results show good tracking performance of both methods using the RNN identifier and the LQR or RL controllers since the non-linear system is simple and the reference is designed to avoid complex solutions. The main difference between the RNN and the RL solution is their accuracy, since our RL takes into account the modelling error then the output control policy is improved. We used the mean error to see the accuracy of each controller as follows:

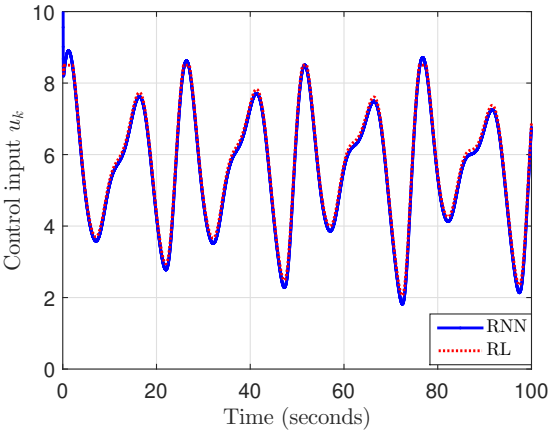
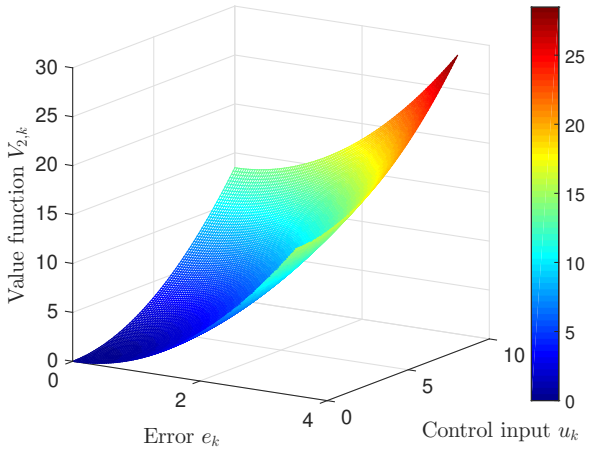


Fig. 2: Tracking control for the surge tank

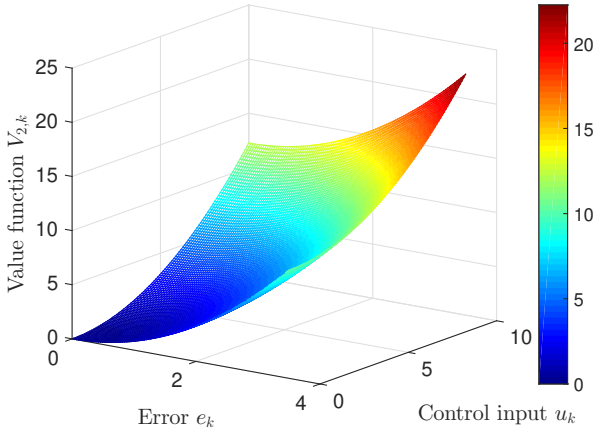
$$\bar{e}_{RNN} = \frac{1}{k}\sum_{i=0}^{k} e_{RNN_i} = 0.0707$$

$$\bar{e}_{RL} = \frac{1}{k}\sum_{i=0}^{k} e_{RL_i} = 0.0032.$$

Notice that the RL error is much smaller than the RNN error. Another advantage of our RL approach is that we enhanced the robustness of the controller in presence of modelling error and converges to an optimal or near optimal control policy. On the other hand, the RNN solution is simple to design but it can only guarantee local optimal performance which is affected by disturbances or modelling error.



(a) RNN learning curve



(b) RL learning curve

Fig. 3: Value function $V_{2,k}$ learning curves

The learning curve of the value function $V_{2,k}$ using RNN and RL is given in Fig. 3. Here is more evident how the modelling error affects the controller design, even more the use of a discounted factor is essential because if the reference trajectory does not go to zero, then the value function is infinite and the term $U_2^\top R_c U_2$ does not go to zero as time goes to infinity.

V. Conclusion

In this work, the discrete-time critic-learning control is proposed. The critic-learning method is based on a serial-parallel recurrent neural network which serves for system identification and tracking control. The tracking control is achieved by using a model-compensation via the neural model and a feedback term. The feedback controller is designed

using two control techniques: discrete LQR control and reinforcement learning. Stability and convergence are presented using Lyapunov stability theory and the contraction property. Simulations are carried out to show that our approach presents optimal or near optimal performances with high accuracy without knowledge of the system dynamics.

Since artificial neural networks have demonstrate the effectiveness as an model identifier, then further work will consider the use of neuromorphic neural networks as approximators of reinforcement learning architectures for the design of optimal and robust controllers.

REFERENCES

[1] F.L. Lewis, D. Vrabie. Optimal Control. *New York, NY, USA: Wiley*, 2012.

[2] F. L. Lewis, D. Vrabie, K. G. Vamvoudakis, Reinforcement Learning and Feedback control using natural decision methods to design optimal adaptive controllers. *IEEE Control Systems Magazine*, 2012.

[3] A. Al-Tamimi, F. L. Lewis, and M. Abu-Khalaf, Discrete-time nonlinear HJB solution using approximate dynamic programming: Convergence proof, *IEEE Transactions on System, Man, and Cybernetics Part B, Cybernetics*, vol. 38(4), pp. 943-949, 2008.

[4] H. Zhang, D. Liu, Y. Luo, and D. Wang. Adaptive Dynamic Programming for Control. *London, U.K.: Springer-Verlag*, 2013.

[5] R.S. Sutton, A.G. Barto, Reinforcement Learning: An Introduction. *Cambridge, MA: MIT Press*, 1998.

[6] H. Modares and F. L. Lewis, Linear quadratic tracking control of partially-unknown continuous-time systems using reinforcement learning, *IEEE Trans. Autom. Control*, vol. 59 (11), pp. 3051-3056, 2014.

[7] B. Kiumarsi and F.L. Lewis. Actor-critic based optimal tracking for partially unknown nonlinear discrete- time systems. *IEEE Transactions on Neural Networks and Learning Systems*, vol. 26(1), pp. 140-151, 2015.

[8] D. Vrabie and F. Lewis, Neural network approach to continuous time direct adaptive optimal control for partially unknown nonlinear systems, *Neural Networks*, vol. 22 (3), pp. 237-246, 2009.

[9] H. Modares and F. L. Lewis, Optimal tracking control of nonlinear partially-unknown constrained-input systems using integral reinforcement learning, *Automatica*, vol. 50 (7), pp. 1780-1792, 2014.

[10] Q. Xie, B. Luo, F. Tan, Discrete-time LQR optimal tracking control problems using Approximate dynamic programming algorithm with Disturbance. *2013 Fourth International Conference on Intelligent Control and Information Processing (ICICIP)*, Beijing China, 2013.

[11] H. Modares, F. L. Lewis, and Z.-P. Jiang, H_∞ tracking control of completely unknown continuous-time systems via off-policy reinforcement learning, *IEEE Transactions on Neural Networks and Learning Systems*, vol. 26(10), pp. 2550-2562, 2015.

[12] B. Kiumarsi, K.G. Vamvoudakis, H. Modares, F.L. Lewis. Optimal and Autonomous control using Reinforcement Learning: A Survey. *IEEE Transactions on Neural Networks and Learning Systems*, vol. 29 (6), 2018.

[13] A. Perrusquía, W. Yu, A. Soria, Position/force control of robot manipulators using reinforcement learning, *Industrial Robot: the international journal of robotics research and application*, vol 46 (2), pp. 267-280, 2019.

[14] I. Grondman, L. Buşoniu, R. Babûska, Model learning actor-critic algorithms: Performance Evaluation in a Motion Control Task. *51st IEEE Conference on Decision and Control*, 2012.

[15] C. Wang, Y. Li, S. S. Ge, T. H. Lee, Optimal Critic Learning for Robot Control in Time-Varying Environments. *IEEE Transactions on Neural Networks and Learning Systems*, vol. 26, no. 10, 2015.

[16] A. Perrusquía, W. Yu, A. Soria, Large space dimension reinforcement learning for robot position/force discrete control, *6th International Conference on Control, Decision and Information Technologies (CoDIT19)*, Paris France, 2019.

[17] K.G. Vamvoudakis, D. Vrabie, F.L. Lewis, Online learning algorithm for zero-sum games with integral Reinforcement Learning. *Journal of Artificial Intelligence and Soft Computing Research*, vol. 11 (4), pp. 315-332, 2011.

[18] K.G. Vamvoudakis, F.L. Lewis, Online actor-critic algorithm to solve the continuous-time infinite horizon optimal control problem. *Automatica*, vol. 46, pp. 878-888, 2010.

[19] D. Vrabie, K. G. Vamvoudakis, and F. L. Lewis. Optimal Adaptive Control and Differential Games by Reinforcement Learning Principles (Control, Robotics and Sensors). *Edison, NJ, USA: IET*, 2013.

[20] I. Grondman, M. Vaandrager, L. Buşoniu, R. Babûska, E. Schuitema, Actor-critic control with reference model learning. *Proc. of the 18th World Congress The International Federation of Automatic Control*, 2011.

[21] A. Perrusquía, W. Yu, X. Li. Impedance Control without environment model by reinforcement learning, *10th International Conference on Intelligent Control and Information Processing (ICICIP 2019)*, Marrakesh, Morocco, 2019.

[22] R. Kamalapurkar, P. Walters, and W. E. Dixon, Model-based reinforcement learning for approximate optimal regulation, *Automatica*, vol. 64, pp. 94-104, 2016.

[23] I. Grondman, M. Vaandrager, L. Buşoniu, R. Babûska, E. Schuitema, Efficient Model Learning Methods for Actor-Critic Control. *IEEE Transactions on Systems, man, and cybernetics. Part B: Cybernetics*, vol 42, no. 3, 2012.

[24] L. Dong, X. Zhong, C. Sun, and H. He. Adaptive event-triggered control based on heuristic dynamic programming for nonlinear discrete-time systems. *IEEE Transactions on Neural Networks and Learning Systems*, vol 28 (7), pp. 1594-1605, 2016.

[25] A. Sahoo, H. Xu and S. Jagannathan, Near optimal event triggered control of nonlinear discrete-time systems using neurodynamic programming. *IEEE Transactions on Neural Networks and Learning Systems*, vol. 27(9), pp. 1801-1815, 2016.

[26] G.J. Leon, I. Lee. Neural network indirect adaptive control with fast learning algorithm. *Neurocomputing*, vol. 12(2-4), pp. 185-199, 1996.

[27] W. Yu, A.S. Poznyak, Indirect adaptive control via parallel dynamic neural networks. *IEEE Proceedings- Control Theory and Applications*, vol. 146 (1), pp. 25-30, 1999.

[28] W. Yu, X. Li. Discrete-time nonlinear system Identification using Recurrent Neural Networks. *Proceedings of the 42nd IEEE Conference on Decision and Control*, Hawai, USA, 2003.

[29] W. Yu. Nonlinear system identification using discrete-time recurrent neural networks with stable learning algorithms. *Information Sciences*, vol. 158, pp. 131-147, 2004.

[30] W. Yu, X. Li. Recurrent fuzzy neural networks for nonlinear system identification. *22nd IEEE International Symposium on Intelligent Control Part of IEEE Multi-conference on Systems and Control*, Singapore, 2007.

[31] W. Yu. Multiple recurrent neural networks for stable adaptive control. *Neurocomputing*, vol. 70, pp. 430-444, 2006.

[32] G. Cybenko. Approximation by superposition of sigmoidal activation function. *Math, Control, Sig. Syst.* vol. 2, pp. 303-314, 1989.

[33] H. Nounou, K.M. Passino. Stable Auto-tuning of Adaptive Fuzzy/Neural controllers for nonlinear discrete-time systems. *IEEE Transactions on Fuzzy Systems*, vol. 12 (1), 2004.

[34] Z.P Jiang, Y. Wang. Input-to-state stability for discrete-time nonlinear systems, *Automatica*. 2001; 37(2): 857-869.

[35] A. Perrusquía, W. Yu. Robust control under worst case uncertainty for unknown nonlinear systems using modified reinforcement learning. *International Journal of Robust and Nonlinear Control*. 2020: 1-17.