

Robust Control of Linear Systems: A Min-Max Reinforcement Learning Formulation

Juan Alejandro Flores-Campos
UPIITA-IPN.
Instituto Politécnico Nacional
Mexico city, Mexico
jaflores@ipn.mx

Adolfo Perrusquía
School of Aerospace, Transport and Manufacturing
Cranfield University
Bedford, UK
Adolfo.Perrusquia-Guzman@cranfield.ac.uk

Abstract—In this paper, an online robust controller based on a min-max reinforcement learning approach for linear systems is discussed. Disturbances are represented by external signals coupled with the control input which are assumed to be bounded within a set of admissible disturbances. The proposed controller implements a min-max approach which realizes a smooth transition between optimal and robust controllers. Lyapunov stability theory is used to assess the stability and boundedness of the min-max robust formulation. A neural reinforcement learning architecture is used to obtain an approximation of the parameters associated to the optimal cost. Simulations are carried out to validate the proposed approach.

I. INTRODUCTION

In the last two decades, optimal/robust control problems have been solved using different techniques [1]. Classic solutions use complete knowledge of the system dynamics to compute off-line the optimal/robust controller, e.g., the linear quadratic regulator (LQR) [2]–[4] also known as H_2 controller, the H_∞ controller, and the hybrid H_2/H_∞ controller (for more details refer to [5]). These controllers are sensitive to modelling uncertainties which directly affect the performance of the closed-loop system.

One of the major issues in solving optimal/robust control problems relies on the assumption that the exact mathematical dynamic model is available [6]. To overcome this issue, adaptive dynamic programming (ADP) [7] and reinforcement learning (RL) [1], [8] methods have been used to compute online the optimal/robust adaptive controller with and without knowledge [9] of the complete system dynamics. The main assumption, for model-based ADP-RL algorithms [10], [11] is knowledge of the dynamics associated to the control input. Model-free ADP-RL algorithms such as Q-learning [12]–[14] offer an easy way to obtain the optimal/robust controller without any knowledge of the system dynamics [15], [16]. Furthermore, RL for robust control solves an H_∞ control problem [17] which finds the worst-case uncertainty that the system can accept to maintain optimal performance. However, most of the exogenous disturbances do not satisfy the H_∞ robust condition and therefore, the states of the systems are ultimately uniformly bounded (UUB) [18].

The stability and convergence of neural RL methods have been studied in previous works [19]–[21]. The results show that the gradient update rule of the neural RL algorithm

exhibits practical stability with exponential convergence under Persistency of Excitation (PE) conditions [22], [23]. Other studies, incorporate recurrent neural networks (RNN) as compensation of the nonlinear dynamics such that the learning time is improved and the control design is reduced to an optimal linear control problem. However, this method requires two steps: 1) identify the nonlinear dynamics using any RNN structure and 2) obtain the optimal controller using a model-based reinforcement learning algorithm [19]. Model-free RL has been used to deal with perturbed systems using function approximators [24], [25] (e.g., k -nearest neighbours, radial basis functions). In these cases, an optimization problem with constraints is solved. Nevertheless, the learning time increases when the number of states and control inputs increases.

Learning from human expertise is becoming a trending topic in different research communities due to its adaptability and robustness to perform specific tasks, especially in video game applications [26]. Human-behaviour learning has been used to control nonlinear systems with known dynamics by solving a Hamilton-Jacobi-Bellman equation. The algorithm uses different cognitive models that emulate specific aspects of human learning (e.g., memory [27], experience [28]–[30], curiosity, the task [31]). The results show that the RL performance is improved under new cognitive models and skills.

The open issue in the current-state-of-the-art is that ADP/RL for robust control does not consider the real disturbance in the design since they are based on a zero-sum-game. To address this gap, we propose a min-max reinforcement learning algorithm that incorporates the upper-bound of the disturbance in the robust control design. Here, the worst-case uncertainty generates a restricted Hamilton-Jacobi-Bellman equation which is nonlinear. A neural RL architecture is used to approximate its solution and compute a robust control policy and the worst-case uncertainty.

II. PROBLEM DEFINITION

Consider the following perturbed linear system

$$\dot{x}(t) = Ax(t) + B[u(t) + w(t)], \quad x(t_0) = x_0, \quad (1)$$

where $x \in \mathbb{R}^n$ defines the state vector, $u \in \mathbb{R}^m$ is the control input, $w \in \mathbb{R}^m$ is an external disturbance that is coupled to the control input, $A \in \mathbb{R}^{n \times n}$ and $B \in \mathbb{R}^{n \times m}$ define the known

dynamics of the linear system, and $x_0 \in \mathbb{R}^n$ is the initial condition. The following assumption is considered

Assumption 2.1: The disturbance vector $w(t)$ is bounded within a set of admissible disturbances $W \subset \mathbb{R}^m$, i.e., $\|w(t)\| \leq \bar{w}$, where $\bar{w}$ is a known upper bound of the external disturbance.

In a standard zero-sum game formulation, the control objective is to optimize the following cost

$$J = \int_t^\infty (x^\top Sx + u^\top Ru - \gamma^2 w^\top w) d\tau, \quad \forall x, u, w, \quad (2)$$

where $S = S^\top \geq 0 \in \mathbb{R}^{n \times n}$, $R = R^\top > 0 \in \mathbb{R}^{m \times m}$, and $\gamma > 0$ define the weight matrices of the cost. The optimal cost is the one that satisfies the following zero-sum differential game

$$J^* = \min_u \max_w \int_t^\infty (x^\top Sx + u^\top Ru - \gamma^2 w^\top w) d\tau. \quad (3)$$

Solving this optimization problem requires the solution of the following Hamilton-Jacobi-Bellman-Isaacs (HJBI) equation

$$\mathcal{H}(x, u^*, w^*, \nabla J^*) = \nabla^\top J^* (Ax + B[u^* + w^*]) + x^\top Sx + u^{*\top} Ru^* - \gamma^2 w^{*\top} w^* = 0, \quad (4)$$

where $\mathcal{H}(\cdot)$ is the Hamiltonian and $\nabla J^* = \frac{\partial J^*}{\partial x} \in \mathbb{R}^n$ is the partial derivative of the cost respect to x . A quadratic solution for (3) is usually assumed, i.e., $J^* = x^\top \mathcal{P}x$ for some kernel matrix $\mathcal{P} = \mathcal{P}^\top > 0 \in \mathbb{R}^{n \times n}$ that satisfies the following Game Algebraic Riccati equation (GARE)

$$A^\top \mathcal{P} + \mathcal{P}A + S - \mathcal{P}BR^{-1}B^\top \mathcal{P} + \frac{1}{\gamma^2} \mathcal{P}BB^\top \mathcal{P} = 0. \quad (5)$$

Then, the optimal control policy and worst-case uncertainty is given by the H_∞ solution

$$u^*(t) = -R^{-1}B^\top \mathcal{P}x(t) \quad (6)$$

$$w^*(t) = \frac{1}{\gamma^2} B^\top \mathcal{P}x(t). \quad (7)$$

RL/ADP algorithms are typically used to derive online optimal/robust control policies through the interaction of the system and the environment. Here, one of the main problems arise by assuming that the real disturbance $w(t)$ fulfils the worst-case uncertainty w^* . In other words, the value of the worst-case uncertainty is based on the values of the proposed matrices S , R , and γ . Hence, different weight matrices give different worst-case uncertainties w^* which do not necessarily cover the real-world disturbance w . In this scenario, we need to test different weight matrices to ensure the optimal control (6) is capable to compensate the disturbance w . This procedure is time-consuming and is not efficient.

A. Main Result

To deal with the problem of previous section, we take a step back and work with a restricted H_2 problem instead of a

H_∞ problem. The new cost (also known as value function) is given by

$$V(x) = \int_t^\infty \xi(x, u) d\tau, \quad (8)$$

where $\xi(x, u) : \mathbb{R}^n \times U \rightarrow \mathbb{R}$ is the reward function which has the following quadratic structure

$$\xi(x, u) = x^\top Sx + u^\top Ru, \quad (9)$$

and $U \subseteq \mathbb{R}^m$ is a subset of admissible control policies. Notice that in contrast to (2), this new formulation is constrained to a set of admissible control policies and not the entire vector space in $\mathbb{R}^m$. The Hamiltonian of the problem subject to (1) and the value function (8) is

$$H(x, u, w, \nabla V) = \nabla^\top V (Ax + B[u + w]) + \xi(x, u), \quad (10)$$

where $\nabla V = \frac{\partial V}{\partial x}$. The optimal value function is defined by

$$V^*(x) = \min_{u \in U} \max_{w \in W} \int_t^\infty (x^\top Sx + u^\top Ru) d\tau, \quad (11)$$

which is the one that satisfies the following restricted Hamilton-Jacobi-Bellman (RHJB) equation

$$H(x, u^*, w^*, \nabla V^*) = \nabla^\top V^* (Ax + B[u^* + w^*]) + \xi(x, u^*) = 0. \quad (12)$$

The above min-max formulation defines a robustified version of the HJB where the disturbances have been incorporated into the solution.

Remark 2.2: The RHJB equation in (12) differs from the HJBI equation in (4). The main difference lies in the optimization of the worst-case uncertainty [32], that is, the HJBI equation finds the optimal worst-case uncertainty w^* which guarantees that the closed-loop system trajectories have an optimal performance. On the other hand, the RHJB adapts the optimal controller to compensate the real uncertainty w within the set of admissible disturbances W to ensure a near-optimal and robust performance. Notice that, in general, $\|w^*\| \leq \|w\| \leq \bar{w}$.

We need first to determine the worst-case uncertainty $w^* \in W$ under this constrained scenario. Considering only the terms associated to w in (10), we obtain that

$$w^* = \operatorname{argmax}_{w \in W} (\nabla^\top V^* B w) = \operatorname{argmax}_{w \in W} \langle B^\top \nabla V^*, w \rangle, \quad (13)$$

where $\langle \cdot, \cdot \rangle$ defines the inner product that, in this case, corresponds to the standard dot product. The Cauchy-Schwarz inequality is used in the right-hand side of (13) as

$$\begin{aligned} \langle B^\top \nabla V^*, w \rangle &\leq \|B^\top \nabla V^*\| \|w\| \\ &\leq \|B^\top \nabla V^*\| \bar{w} \end{aligned} \quad (14)$$

The inequality (14) is used to determine the worst-case uncertainty w^* by considering only the equality part. The worst-case uncertainty w^* is given by

$$w^* = \frac{\bar{w}}{\|B^\top \nabla V^*\|} B^\top \nabla V^*, \quad (15)$$

Notice that we obtain a normalized worst-case uncertainty in terms of the known upper bound $\bar{w}$ of the disturbance vector w . We add +1 into the denominator of the above equation to avoid any computational issue

$$w^* = \frac{\bar{w}}{\|B^\top \nabla V^*\| + 1} B^\top \nabla V^*. \quad (16)$$

The robust control policy u^* is obtained by differentiating the Hamiltonian (10) with respect to u as

$$u^* = -\frac{1}{2} R^{-1} B^\top \nabla V^*. \quad (17)$$

Substituting the worst-case uncertainty w^* (16) and robust control (17) in the RHJB equation (12) gives

$$\begin{aligned} \nabla^\top V^* A x - \frac{1}{4} \nabla^\top V^* B R^{-1} B^\top \nabla V^* + x^\top S x \\ + \frac{\bar{w}}{\|B^\top \nabla V^*\| + 1} \nabla^\top V^* B B^\top \nabla V^* = 0. \end{aligned} \quad (18)$$

Notice that if we assume that the solution of the RHJB is quadratic, i.e., $V^* = x^\top P x$ for some constant kernel matrix $P = P^\top > 0 \in \mathbb{R}^{n \times n}$, then substituting in (18) yields the restricted algebraic Riccati equation (RARE) given by

$$A^\top P + P A + S - P B R^{-1} B^\top P + \tau P B B^\top P = 0, \quad (19)$$

where $\tau = \frac{4\bar{w}}{\min_x \|2B^\top P x\| + 1}$. Notice that the RARE (19) has an equivalent structure to the GARE (5) where the constant attenuation factor $1/\gamma^2$ is modified to a bound factor τ which adapts the value of the kernel matrix P and thus the robust control policy u^* .

In general, the solution of the RHJB equation cannot be obtained in closed form due to the nonlinear form of (18). In the following section, the solution of the RHJB equation is discussed using reinforcement learning.

III. APPROXIMATE SOLUTION OF THE RHJB EQUATION

In this section, an approximate solution for the RHJB equation is given using a neural reinforcement learning model.

Let the optimal value function (11) be equivalently approximated [33] by

$$V^*(x) = x^\top P_0 x + \phi^\top \theta + \varepsilon_V, \quad (20)$$

where $P_0 = P_0^\top > 0 \in \mathbb{R}^{n \times n}$ is a positive definite kernel matrix, $\phi = \phi(x) : \mathbb{R}^n \rightarrow \mathbb{R}^p$ is a vector of basis or activation functions, $\theta \in \mathbb{R}^p$ denotes the weights of the neural network and $\varepsilon_V = \varepsilon_V(x) : \mathbb{R}^n \rightarrow \mathbb{R}$ is an approximation error. On the one hand, the first element of the right-hand side of the equality in (20) denotes a prior solution of the restricted optimization problem. On the other hand, the second term is designed to learn from interaction using a neural reinforcement learning approximation. The combination of these terms helps the design of a robust controller that is capable to stabilize the linear system and compensate the external disturbance and model uncertainties.

Considering a fixed policy u and disturbance w , the Hamiltonian in terms of the approximation satisfies

$$\begin{aligned} H(x, u, w, P_0, \nabla \phi) = & (2x^\top P_0 + \theta^\top \nabla \phi) (A x + B[u + w]) \\ & + x^\top S x + u^\top R u = \varepsilon_H \end{aligned} \quad (21)$$

where $\varepsilon_H = -\nabla \varepsilon_V^\top \dot{x}$ is the residual error due to the approximation error. Therefore, the robust control policy is given by

$$u^* = -\frac{1}{2} R^{-1} B^\top (2P_0 x + \nabla^\top \phi \theta) \quad (22)$$

and the worst-case uncertainty (16) is given by

$$w^* = \beta B^\top (2P_0 x + \nabla^\top \phi \theta). \quad (23)$$

where $\beta = \beta(x) = \frac{\bar{w}}{\|2P_0 x + \nabla^\top \phi \theta\| + 1}$. Substituting the worst-case uncertainty (23), and the robust control policy (22) in the Hamiltonian (21) gives the following RHJB equation

$$\begin{aligned} x^\top [P_0 A + A^\top P_0 + S - P_0 (B R^{-1} B^\top - 4\beta B B^\top) P_0] x \\ + \theta^\top \nabla \phi (A - B R^{-1} B^\top P_0 + 4\beta B B^\top P_0) x \\ + \theta^\top \nabla \phi \left(-\frac{1}{4} B R^{-1} B^\top + \beta B B^\top \right) \nabla^\top \phi \theta = \varepsilon_H, \end{aligned} \quad (24)$$

For stability analysis purposes, consider an equivalent notation of $\nabla^\top \phi \theta \equiv \Psi \Xi$ where $\Psi \in \mathbb{R}^{n \times q}$ is a matrix composed of the values of θ and the coefficients of $\nabla \phi$, $\Xi \in \mathbb{R}^q$ is given by the unique functions of x in $\nabla \phi$. Then, the following theorem establishes the uniform ultimate boundedness (UUB) of the linear trajectories under the robust control policy (22) and worst-case uncertainty (23).

Theorem 3.1: Consider the linear system (1) under the robust control policy (22) and worst-case uncertainty (23). If the Schur complement conditions

$$\begin{aligned} M_3 &> 0 \\ M_1 - M_2 M_3^{-1} M_2 &> 0, \end{aligned} \quad (25)$$

are satisfied with

$$\begin{aligned} M_1 &= S + P_0 \left[(B R^{-1} B^\top - \Lambda_1) + 4\beta B (1 - \beta \Lambda^{-1}) B^\top \right] P_0 \\ M_2 &= 2\beta \Psi^\top B (I - \beta \Lambda^{-1}) B^\top P_0 + \frac{1}{2} \Psi^\top B R^{-1} B^\top P_0 \\ M_3 &= \Psi^\top \left(\frac{1}{4} B R^{-1} B^\top + \beta B (I - \beta \Lambda^{-1}) B^\top - \Lambda_2 \right) \Psi, \end{aligned}$$

where $\Lambda_1 = \Lambda_1^\top > 0 \in \mathbb{R}^{n \times n}$, $\Lambda_2 = \Lambda_2^\top > 0 \in \mathbb{R}^{n \times n}$ are known positive definite matrices and $\Lambda^{-1} = \Lambda_1^{-1} + \Lambda_2^{-1}$. If there exists a scalar $k = \lambda_{\min}(Q) + \rho$ for a positive scalar $\rho > 0$, then the closed-loop trajectories are UUB with a practical bound

$$\mu = \sqrt{\frac{\|\varepsilon_H\|}{k}}, \quad Q = \begin{bmatrix} M_1 & M_2 \\ M_2^\top & M_3 \end{bmatrix} \in \mathbb{R}^{(n+q) \times (n+q)}. \quad (26)$$

Proof 1: Consider the following Lyapunov function

$$V = x^\top P_0 x + \phi^\top \theta, \quad (27)$$

which satisfies the Rayleigh condition

$$\lambda_{\min}(P_0) \|x\|^2 \leq V \leq \lambda_{\max}(P) \|x\|^2 + \theta^\top \phi. \quad (28)$$

The above inequality means that the value function is bounded by class- $\mathcal{K}$ functions [34]. The time-derivative of V along the linear system trajectories (1) is

$$\begin{aligned}\dot{V} = & x^\top (P_0 A + A^\top P - 2P_0 B R^{-1} B^\top P_0) x + 2x^\top P_0 B w \\ & - 2x^\top P_0 B R^{-1} B^\top \nabla^\top \phi \theta + \theta^\top \nabla \phi B w \\ & + \theta^\top \nabla \phi A x - \frac{1}{2} \theta^\top \nabla \phi B R^{-1} B^\top \nabla \phi \theta.\end{aligned}\quad (29)$$

The Young's inequality is applied in the terms associated to the disturbance w as

$$\begin{aligned}\dot{V} \leq & x^\top (P_0 A + A^\top P - 2P_0 B R^{-1} B^\top P_0) x + \theta^\top \nabla \phi A x \\ & - 2x^\top P_0 B R^{-1} B^\top \nabla^\top \phi \theta - \frac{1}{2} \theta^\top \nabla \phi B R^{-1} B^\top \nabla \phi \theta \\ & + x^\top P_0 \Lambda_1 P_0 x + w^\top B^\top \Lambda_1^{-1} B w + \theta^\top \nabla \phi \Lambda_2 \nabla^\top \phi \theta \\ & + w^\top B^\top \Lambda_2^{-1} B w \\ = & x^\top (P_0 A + A^\top P - 2P_0 B R^{-1} B^\top P_0 + P_0 \Lambda_1 P_0) x \\ & + w^\top B^\top \Lambda^{-1} B w + \theta^\top \nabla \phi \left(A x - 2B R^{-1} B^\top P_0 x \right. \\ & \left. - \frac{1}{2} B R^{-1} B^\top \nabla \phi \theta + \Lambda_2 \nabla^\top \phi \theta \right).\end{aligned}\quad (30)$$

Substituting the RHJB (24) and worst-case uncertainty (16) in (30) gives

$$\begin{aligned}\dot{V} \leq & -\varepsilon_H - x^\top \left(S + P_0 (B R^{-1} B^\top - \Lambda_1) P_0 \right. \\ & \left. + 4\beta P_0 B (I - \beta \Lambda^{-1}) B^\top P_0 \right) x - \Xi^\top \Psi^\top \left(\frac{1}{4} B R^{-1} B^\top \right. \\ & \left. + \beta B (I - \beta \Lambda^{-1}) B^\top - \Lambda_2 \right) \Psi \Xi \\ & - \Xi^\top \Psi^\top \left(4\beta B (I - \beta \Lambda^{-1}) B^\top P_0 + B R^{-1} B^\top P_0 \right) x \\ = & - \begin{bmatrix} x \\ \Xi \end{bmatrix}^\top \begin{bmatrix} M_1 & M_2 \\ M_2^\top & M_3 \end{bmatrix} \begin{bmatrix} x \\ \Xi \end{bmatrix} - \varepsilon_H = -X^\top Q X - \varepsilon_H \\ \leq & -\lambda_{\min}(Q) \|X\|^2 + \|\varepsilon_H\|,\end{aligned}\quad (31)$$

where $X = [x^\top, \Xi^\top]^\top \in \mathbb{R}^{n+q}$. If we select a large enough matrix S and appropriate matrices Λ_1 and Λ_2 such as the Schur complement conditions in Theorem 3.1 are satisfied, then Q is positive definite. Furthermore, for a large enough S it implies that β is small constant such that $(I - \beta \Lambda^{-1}) > 0$. The Lyapunov function candidate is negative definite if

$$\|X\| > \sqrt{\frac{\|\varepsilon_H\|}{\lambda_{\min}(Q) + \rho}} \equiv \mu, \quad (32)$$

then there exists a scalar $k > 0$ that ensures that X remains within a bounded set S_μ of radius μ , i.e., $\|X\| \leq \mu$ and hence, the trajectories are UUB. This completes the proof.

The weights θ of the NN and the kernel matrix P_0 that provide the best approximation of (20) are unknown. Therefore, the following value function approximation is used

$$\hat{V} = x^\top \hat{P}_0 x + \hat{\theta}^\top \phi \quad (33)$$

where $\hat{\theta} \in \mathbb{R}^p$ are the estimates of the neural weights θ , and $\hat{P}_0 = \hat{P}^\top > 0 \in \mathbb{R}^{n \times n}$ is an estimate of the kernel matrix P_0 . The design of the complete neural reinforcement learning is discussed in the next section.

A. Neural Reinforcement Learning

One might think that the most effective method to calculate (online) the kernel matrix $\hat{P}_0$ is by means of a differential Ricatti equation (DRE) of the form

$$\dot{\hat{P}}_0 + A^\top \hat{P}_0 + \hat{P}_0 A + S - \hat{P}_0 B R^{-1} B^\top \hat{P}_0 = 0. \quad (34)$$

The DRE (34) does not contain information of the external disturbance vector w . Therefore, the solution $\hat{P}_0$ lacks of robustness and adaptability, but serves to stabilize the model.

Remark 3.2: The DRE in (34) is in fact an Algebraic Ricatti equation (ARE) because the kernel matrix $\hat{P}_0$ is constant.

To solve the robustness problem of the DRE, let first rewrite the right hand side of (33) in the next equivalent form

$$x^\top \hat{P}_0 x \equiv \text{vech}(\hat{P}_0)^\top (x \otimes x). \quad (35)$$

where $\otimes$ is symmetric Kronecker product and $\text{vech}(\hat{P})$ is the half-vectorization of matrix $\hat{P}$. Now consider the neural reinforcement learning element. Here, the value function (33) is rewritten in terms of (35) as

$$\hat{V} = \text{vech}(\hat{P}_0)^\top (x \otimes x) + \hat{\theta}^\top \phi = \hat{\Theta}^\top \Phi, \quad (36)$$

where $\Phi = [(x \otimes x)^\top, \phi^\top]^\top \in \mathbb{R}^{\frac{n(n+1)}{2}+p}$ is the vector of basis functions and $\hat{\Theta} = [\text{vech}(\hat{P}_0)^\top, \hat{\theta}^\top]^\top \in \mathbb{R}^{\frac{n(n+1)}{2}+p}$ is the vector of parameter estimates.

Remark 3.3: The elements of the basis function $\phi = [\phi_1, \dots, \phi_p]^\top$ must be different to the basis functions within $(x \otimes x)$ to avoid linear dependence.

The RHJB (24) in terms of the value function approximation (36) gives as output a temporal difference (TD) error [11] described by the next equality

$$\hat{\Theta}^\top \nabla \Phi (A x + B[u + w]) + x^\top S x + u^\top R u = \delta. \quad (37)$$

The main goal of the neural reinforcement learning algorithm is to minimize the squared of the TD error. The gradient-descent rule proposed in [35] is used to compute the parameter estimates as

$$\dot{\hat{\Theta}} = -\alpha \frac{q}{(1 + q^\top q)^2} \left(q^\top \hat{\Theta} + x^\top S x + u^\top R u \right), \quad (38)$$

where $\alpha > 0$ is the learning rate and $q = \nabla \Phi (A x + B[u + w])$.

Convergence of the estimates is achieved under the fulfilment of a persistency of excitation (PE) condition [22]. So, the robust decision making controller and the worst-case uncertainty are given by

$$u^* = -\frac{1}{2} R^{-1} B^\top \nabla^\top \Phi \hat{\Theta}. \quad (39)$$

$$w^* = \bar{w} \frac{B^\top \nabla^\top \Phi \hat{\Theta}}{\|B^\top \nabla^\top \Phi \hat{\Theta}\| + 1}. \quad (40)$$

The boundedness of the closed-loop trajectories of the neural reinforcement learning is given in [11]. Here it is shown that the parameters of the network converges exponentially to a compact set ν whose radius is proportional to the residual error ε_H as $t \rightarrow \infty$.

IV. SIMULATION RESULTS

In this section, the effectiveness of the proposed min-max reinforcement learning formulation is verified with the following case study.

Consider the F16 aircraft plant used in [35]

$$\begin{aligned}\dot{x} &= Ax + B(u + w) \\ &= \begin{bmatrix} -1.01887 & 0.90506 & -0.00215 \\ 0.82225 & -1.07741 & -0.17555 \\ 0 & 0 & -1 \end{bmatrix} x + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} u \\ &\quad + [0 \ 0 \ 1]^\top 0.19 \sin(t).\end{aligned}$$

In this paper, several weight matrices are tested until the best performance is achieved. The final weight matrices are set to $S = 50I$ and $R = 1$. We compare three different approaches: LQR, the synchronous neural RL (SNRL) [35], and the proposed min-max RL (MMRL). Both the LQR and SNRL are formulated in a H_2 sense, that is, without considering the worst-case uncertainty w^* in the control design to avoid biased results. The external disturbance w satisfies the following bound $|w| \leq 0.2 = \bar{w}$. The control objective is to stabilize the aircraft's dynamics at the origin.

The exact kernel solution P_0 of the LQR is

$$P_0 = \begin{bmatrix} 68.1705 & 55.1946 & -1.3616 \\ 55.1946 & 68.3563 & -1.6166 \\ -1.3616 & -1.6166 & 6.1815 \end{bmatrix}.$$

Six basis functions $\Phi = [x_1^2, x_1x_2, x_1x_3, x_2^2, x_2x_3, x_3^2]^\top$ are considered in the SNRL. A sum of exponential weighted sinusoidal functions with different frequencies is used as PE signal to guarantee parameter estimates convergence. The learning rate is set to $\alpha = 52$ which is the one that obtains better results without destabilizing the gradient algorithm. The final kernel solution $\hat{P}_0$ using the SNRL is

$$\hat{P}_0 = \begin{bmatrix} 70.9717 & 52.5037 & -1.391 \\ 52.5037 & 70.8515 & -1.5779 \\ -1.391 & -1.5779 & 6.1829 \end{bmatrix}.$$

For the proposed MMRL, seven basis functions are used $\Phi = [x_1^2, x_1x_2, x_1x_3, x_2^2, x_2x_3, x_3^2, \tanh(x^\top x)]^\top$. We use the same learning rate and PE signal as in the SNRL. Fig. 1 shows the parameter estimates convergence of the proposed MMRL.

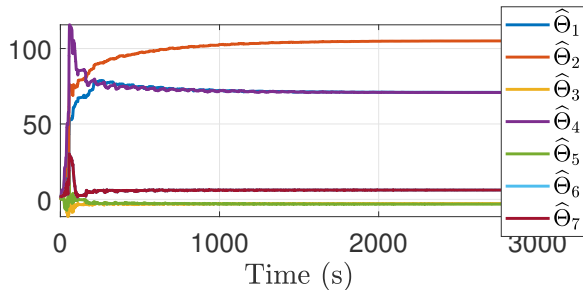


Fig. 1. Parameter estimates convergence using the proposed MMRL.

$$\hat{P}_0 = \begin{bmatrix} \hat{\Theta}_1 & \frac{1}{2}\hat{\Theta}_2 & \frac{1}{2}\hat{\Theta}_3 \\ \frac{1}{2}\hat{\Theta}_2 & \hat{\Theta}_4 & \frac{1}{2}\hat{\Theta}_5 \\ \frac{1}{2}\hat{\Theta}_3 & \frac{1}{2}\hat{\Theta}_5 & \hat{\Theta}_6 \end{bmatrix} = \begin{bmatrix} 68.46 & 55.5 & -2.105 \\ 55.5 & 68.7 & -2.462 \\ -2.105 & -2.462 & 9.014 \end{bmatrix},$$

The parameters converge to the following values and $\hat{\Theta}_7 = 7.8$. So, each control policy is given by

$$u_{LQR} = -R^{-1}B^\top P_0 x,$$

$$u_{SNRL} = -\frac{1}{2}[\hat{\Theta}_3 x_1 + \hat{\Theta}_5 x_2 + 2\hat{\Theta}_6 x_3],$$

$$u_{MMRL} = -\frac{1}{2}[\hat{\Theta}_3 x_1 + \hat{\Theta}_5 x_2 + 2x_3(\hat{\Theta}_6 + \hat{\Theta}_7 \text{sech}^2(x^\top x))].$$

Fig. 2 shows the stabilization of the F16 states. Notice that the LQR and SNRL have similar results where they attenuate the effect of the external disturbance due to the selection of matrix S . On the other hand, the proposed approach exhibits more attenuation of the disturbance. To exhibit the robust performance of the proposed approach, we test it under two different scenarios. The first experiment uses $w = 0$ and the second one uses a disturbance of $w = 0.1c_{5t} + 0.1s_{3t}$, where $c_x = \cos(x)$ and $s_x = \sin(x)$. Table I summarizes the Euclidean norm $\|x\|$ results of the last 45 seconds.

TABLE I
EUCLIDEAN NORM OF SYSTEM TRAJECTORIES WITH/WITHOUT DISTURBANCES

		LQR	SNRL	MMRL
$w = 0$	$\ x\ $	0.002	0.002	0.0027
	$\ u\ $	$4.39e-4$	$4.52e-4$	$3.36e-4$
$w = 0.1(c_{5t} + s_{3t})$	$\ x\ $	0.0125	0.0125	0.0061
	$\ u\ $	0.0764	0.0764	0.0932

The obtained results are consistent with our claims. Here, it is shown that the proposed approach can attenuate the disturbances in a better way in comparison to standard H_2 controllers. However, as expected, the control input is greater since the robust policy considers the worst-case uncertainty w^* . Equivalently, when there is no disturbance, the proposed approach slightly degrades the optimal control performance which can be regarded as a compromise in the transition between robust and optimal control.

V. CONCLUSIONS

In this paper, the design of robust controllers for linear systems is formulated in terms of a min-max reinforcement learning approach. The proposed approach bridges the optimal and robust control theories by restricting both the control policy and worst-case uncertainty into admissible sets instead of considering the whole vector space. A restricted HJB equation is designed in a min-max form which facilitates the incorporation of the external disturbances in the robust control design. Lyapunov stability theory is used to assess the stability and boundedness of the proposed formulation. A neural reinforcement learning algorithm is used as the main tool to estimate the optimal parameters' values associated to the optimal cost. The results show improvement in the disturbance attenuation whilst maintaining a robust performance.

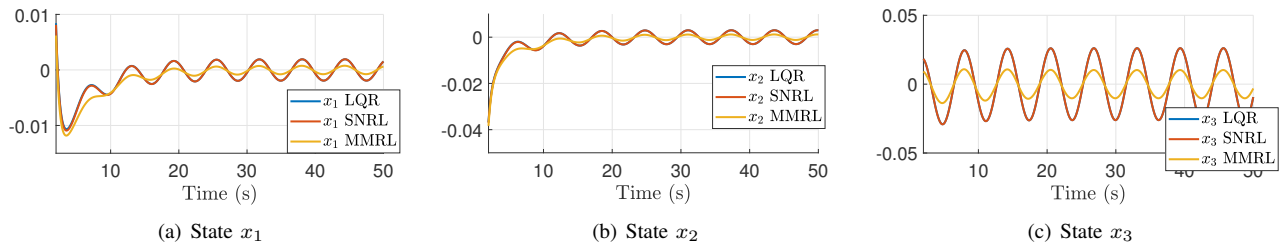


Fig. 2. Stabilization of the F16 states

This approach can be easily expanded to unknown dynamics using an integral reinforcement learning formulation. Further work will focus the extension of the approach for nonlinear systems and time-varying systems.

REFERENCES

- [1] B. Kiumarsi, G. V. Kyriakos, H. Modares, and F. L. Lewis, "Optimal and autonomous control using reinforcement learning: A survey," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 29, no. 6, pp. 2042–2062, 2018.
- [2] S. A. A. Rizvi and Z. Lin, "Output feedback q-learning control for the discrete-time linear quadratic regulator problem," *IEEE transactions on neural networks and learning systems*, vol. 30, no. 5, pp. 1523–1536, 2018.
- [3] A. Perrusquía, "Solution of the linear quadratic regulator problem of black box linear systems using reinforcement learning," *Information Sciences*, vol. 595, pp. 364–377, 2022.
- [4] M. Ferizbegovic, J. Umenberger, H. Hjalmarsson, and T. B. Schön, "Learning robust lq-controllers using application oriented exploration," *IEEE Control Systems Letters*, vol. 4, no. 1, pp. 19–24, 2019.
- [5] A. Perrusquía and W. Yu, "Continuous-time reinforcement learning for robust control under worst-case uncertainty," *International Journal of Systems Science*, vol. 52, no. 4, pp. 770–784, 2021.
- [6] J. A. Flores-Campos, A. Perrusquía, L. H. Hernández-Gómez, N. González, and A. Armenta-Molina, "Constant speed control of slider-crank mechanisms: A joint-task space hybrid control approach," *IEEE Access*, vol. 9, pp. 65676–65687, 2021.
- [7] Y. Jiang and Z.-P. Jiang, "Computational adaptive optimal control for continuous-time linear systems with completely unknown dynamics," *Automatica*, vol. 48, no. 10, pp. 2699–2704, 2012.
- [8] R. Sutton and A. Barto, *Reinforcement Learning: An Introduction*. Cambridge, MA: MIT Press, 1998.
- [9] H. Xu, S. Jagannathan, and F. L. Lewis, "Stochastic optimal control of unknown linear networked control system in the presence of random delays and packet losses," *Automatica*, vol. 48, no. 6, pp. 1017–1030, 2012.
- [10] H. Modares and F. L. Lewis, "Linear quadratic tracking control of partially-unknown continuous-time systems using reinforcement learning," *IEEE Transactions on Automatic Control*, vol. 59, no. 11, pp. 3051–3056, 2014.
- [11] A. Perrusquía and W. Yu, "Neural $\mathcal{H}_2$ control using continuous-time reinforcement learning," *IEEE Transactions on Cybernetics*, vol. 52, no. 6, pp. 4485–4494, 2022.
- [12] B. Kiumarsi, F. L. Lewis, H. Modares, A. Karimpour, and M.-B. Naghibi-Sistani, "Reinforcement Q-learning for optimal tracking control of linear discrete-time systems with unknown dynamics," *Automatica*, vol. 50, no. 4, pp. 1167–1175, 2014.
- [13] K. G. Vamvoudakis, "Q-learning for continuous-time linear systems: A model-free infinite horizon optimal control approach," *Systems & Control Letters*, vol. 100, pp. 14–20, 2017.
- [14] M. Palanisamy, H. Modares, F. L. Lewis, and M. Aurangzeb, "Continuous-time Q-learning for infinite-horizon discounted cost linear quadratic regulator problems," *IEEE Transactions on Cybernetics*, vol. 45, no. 2, pp. 165–176, 2015.
- [15] L. M. Zhu, H. Modares, G. O. Peen, F. L. Lewis, and B. Yue, "Adaptive suboptimal output-feedback control for linear systems using integral reinforcement learning," *IEEE Transactions on Control Systems Technology*, vol. 23, no. 1, pp. 264–273, 2015.
- [16] M. Li, J. Qin, W. X. Zheng, Y. Wang, and Y. Kang, "Model-free design of stochastic lqr controller from a primal–dual optimization perspective," *Automatica*, vol. 140, p. 110253, 2022.
- [17] B. Luo, H.-N. Wu, Huand-Tingwen, and D. Liu, "Reinforcement learning solution for HJB equation arising in constrained optimal control problem," *Neural Networks*, vol. 71, pp. 150–158, 2015.
- [18] A. Perrusquía and W. Guo, "Physics informed trajectory inference of a class of nonlinear systems using a closed-loop output error technique," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 2023.
- [19] K. Vamvoudakis, D. Vrabie, and F. L. Lewis, "Online policy iteration based algorithms to solve the continuous-time infinite horizon optimal control problem," *2009 IEEE Symposium on Adaptive Dynamic Programming and Reinforcement Learning*, 2009.
- [20] A. Perrusquía and W. Yu, "Discrete-time $\mathcal{H}_2$ neural control using reinforcement learning," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 32, no. 11, pp. 4879–4889, 2021.
- [21] D. Vrabie and F. L. Lewis, "Neural network approach to continuous-time direct adaptive optimal control for partially unknown nonlinear systems," *Neural Networks*, vol. 22, pp. 237–246, 2009.
- [22] A. Perrusquía and W. Yu, "Identification and optimal control of nonlinear systems using recurrent neural networks and reinforcement learning: An overview," *Neurocomputing*, vol. 438, pp. 145–154, 2021.
- [23] F. Lewis, S. Jagannathan, and A. Yeşildirek, *Neural Network control of robot manipulators and nonlinear systems*. Taylor & Francis, 1999.
- [24] L. Busoniu, R. Babuska, B. De Schutter, and D. Ernst, *Reinforcement learning and dynamic programming using function approximators*, vol. 39. CRC press, 2010.
- [25] W. Yu and A. Perrusquía, *Human-Robot Interaction Control Using Reinforcement Learning*. John Wiley & Sons, 2021.
- [26] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, et al., "Human-level control through deep reinforcement learning," *nature*, vol. 518, no. 7540, pp. 529–533, 2015.
- [27] H. Modares, F. Lewis, and M.-B. Naghibi-Sistani, "Integral reinforcement learning and experience replay for adaptive optimal control of partially-unknown constrained-input continuous-time systems," *Automatica*, vol. 50, pp. 193–202, 2014.
- [28] J. Ramírez, W. Yu, and A. Perrusquía, "Model-free reinforcement learning from expert demonstrations: a survey," *Artificial Intelligence Review*, vol. 55, no. 4, pp. 3213–3241, 2022.
- [29] E. Pignat, J. Silvério, and S. Calinon, "Learning from demonstration using products of experts: applications to manipulation and task prioritization," *arXiv preprint arXiv:2010.03505*, 2020.
- [30] A. Perrusquía and W. Guo, "Optimal control of nonlinear systems using experience inference human-behavior learning," *IEEE/CAA Journal of Automatica Sinica*, vol. 10, no. 1, pp. 90–102, 2023.
- [31] A. Perrusquía, W. Yu, and X. Li, "Nonlinear control using human behavior learning," *Information Sciences*, vol. 569, pp. 358–375, 2021.
- [32] A. Perrusquía and W. Yu, "Robust control under worst-case uncertainty for unknown nonlinear systems using modified reinforcement learning," *International Journal of Robust and Nonlinear Control*, vol. 30, no. 7, pp. 2920–2936, 2020.
- [33] J. N. Hendriks, J. R. Holdsworth, A. G. Wills, T. B. Schön, and B. Ninness, "Data to controller for nonlinear systems: An approximate solution," *IEEE Control Systems Letters*, vol. 6, pp. 1196–1201, 2021.
- [34] H. Khalil, *Nonlinear systems*. Prentice Hall, 2002.
- [35] K. G. Vamvoudakis and F. L. Lewis, "Online actor–critic algorithm to solve the continuous-time infinite horizon optimal control problem," *Automatica*, vol. 46, no. 5, pp. 878–888, 2010.



DSPACE

<https://dspace.org/>

Robust control of linear systems: a min-max reinforcement learning formulation

Flores-Campos, Juan Alejandro; Perrusquía, Adolfo

2023-12-05

Institute of Electrical and Electronics Engineers (IEEE)

<https://doi.org/10.1109/CCE60043.2023.10332826>

Flores-Campos JA, Perrusquía A. (2023) Robust control of linear systems: a min-max reinforcement learning formulation. In: 2023 20th International Conference on Electrical Engineering, Computing Science and Automatic Control (CCE), 25-27 October 2023, Mexico City, Mexico