Information Sciences 595 (2022) 364–377

Contents lists available at ScienceDirect

Information Sciences

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

Solution of the linear quadratic regulator problem of black box linear systems using reinforcement learning

Adolfo Perrusquía

School of Aerospace, Transport and Manufacturing, Cranfield University, Bedford MK43 0AL, UK

ARTICLE INFO

Article history:
Received 24 May 2021
Received in revised form 22 September 2021
Accepted 1 March 2022
Available online 5 March 2022

Keywords:
Linear quadratic regulator
State observer parametrization
Q-learning
Gradient descent
Output feedback
Persistency of excitation

ABSTRACT

In this paper, a Q-learning algorithm is proposed to solve the linear quadratic regulator problem of black box linear systems. The algorithm only has access to input and output measurements. A Luenberger observer parametrization is constructed using the control input and a new output obtained from a factorization of the utility function. An integral reinforcement learning approach is used to develop the Q-learning approximator structure. A gradient descent update rule is used to estimate on-line the parameters of the Q-function. Stability and convergence of the Q-learning algorithm under the Luenberger observer parametrization is assessed using Lyapunov stability theory. Simulation studies are carried out to verify the proposed approach.

1. Introduction

Black box systems are commonly used to model systems which can be viewed in terms of measurements of its inputs and outputs [1]. For these kind of systems, both the internal states and the parameters of the system are unknown [2,3], e.g., robots [4,5], unmanned vehicles [6], chemical and biological processes [7], among others.

Control of black box systems have been well studied using model-free (e.g., PD, PID, sliding mode control, etc.)[8,9] and adaptive (e.g., neural networks, fuzzy systems, indirect and direct adaptive algorithms, etc.) controllers [10,5]. However, the above controllers cannot ensure optimal performances [11]. Furthermore, they need access to the internal states which can only be achieved by means of a state estimator.

In the sequel of this paper, black box linear systems are considered. For this kind of systems, state observers are used to estimate the internal states [12]. However, they need knowledge of the parameters of the system which are unknown. For certain black box systems such as mechanical systems, sliding mode differentiators (e.g., the Levant's differentiator [13]) can be used to estimate the derivatives of the output. Nevertheless, the degree of the closed-loop system and the number of control gains are increased [14]. In consequence, it makes harder the stability analysis of the closed-loop system between the black box system and the model-free controller design.

Reinforcement learning (RL) [15] is a machine learning technique that merges the main advantages of adaptive and optimal control theories[16]. In the context of control theory, the RL algorithms that achieve both optimal and adaptive performances are called Adaptive Dynamic Programming (ADP)[17,18] algorithms. They seek on-line the solution of a Hamilton–

E-mail address: Adolfo.Perrusquia-Guzman@cranfield.ac.uk

https://doi.org/10.1016/j.ins.2022.03.004
0020-0255/

Jacobi-Bellman (HJB) equation[19,20]. In particular, for linear systems the HJB equation boils down to finding the solution of an Algebraic Riccati equation (ARE) which, in this case, is known as the linear quadratic regulator (LQR) problem [21,22].

It is well known that the LQR problem is a model-based dynamic programming algorithm and one way to solve it (without system information) is by means of model-free RL/ADP algorithms [23,24]. In the literature, most of these algorithms have an actor-critic structure [25] and use neural networks as approximators [26–28] combined with other techniques such as experience replay or eligibility traces [29–31]. Their success rely in a persistency of excitation (PE) condition fulfilment [32], whose function is to excite the system modes such that the estimates of the ARE solution converge to their real values.

Some of the most famous model-free RL/ADP algorithms are Lyapunov recursions [17], critic algorithms [33], actor-critic [34,35], and Q-learning [36,27]. These methods find the solution of the ARE without knowledge of the system dynamics, however they require state's measurements which cannot be estimated by a state observer due to the misknowledge of the black box system's parameters.

1.1. Related work

To overcome the above issues, in [37] an output-feedback control was proposed to use output measurements to compute the optimal control law. The main issue of this approach is that it requires that the rank of the output matrix must be equal to the number of the system's states to guarantee complete state feedback. In [38], a state parametrization is used to formulate a Q-learning algorithm for discrete-time systems. This algorithm finds recursively the solution of the ARE via an off-line least squares (LS) policy iteration (PI) algorithm. However, the approach assumes that the state estimation is equal to the real state. This assumption is not satisfied in the short term due to the incorporation of a state estimation error which fades as time increases. Furthermore, the new value function parametrization is not a kernel matrix. This fact is discussed in future sections.

In summary, the main concerns and motivation of this work are: (i) output-feedback controllers need a full rank output matrix which is not common in real applications, (ii) the use of state estimators require to be analysed in the closed-loop system and not assume that both signals are equivalent, (iii) there is no theoretical proof of the model-free RL under the state estimator.

In this paper, a solution of the LQR problem of black box linear systems is proposed. In contrast to previous works, the proposed approach obtains the solution of the LQR problem without knowledge of the internal states of the black box system using only measurements of the input, output, and the state observer parameterization signals. Furthermore, a new output is proposed to correlate the estimation and control problems in terms of a factored utility function that avoids the full rank output matrix assumption. The optimal controller is formulated with a new parameterization of the Q-function to take into account the new states of the state observer parameterization. Then, a Q-learning algorithm based on the gradient descent technique and inverse reinforcement learning formulation is used to obtain the optimal control policy without knowledge of the system dynamics. Rigorous stability analysis using Lyapunov stability theory are given to support the proposed technique. Simulations are carried out to verify the complete approach.

1.2. Contributions and notation

The main contributions are: (i) a state observer parameterization in terms of the input measurements and a new output which is obtained from a factorization of the utility function, (ii) a new Q-function parameterization that takes into account the states of the state-observer parameterization, (iii) the integral reinforcement learning and the gradient descent methodologies are used to estimate the optimal Q-function and the optimal control law, (iv) a stability analysis of both the state parameterization and the Q-learning algorithm are provided to verify the uniqueness of the optimal controller and that the closed-loop trajectories are bounded and converge exponentially to zero. Therefore, the optimal controller is obtained without internal states measurements and parameters information of the black box system.

Throughout this paper, $\mathbb{N},\ \mathbb{R},\ \mathbb{R}_+,\ \mathbb{R}^n,\ \mathbb{R}^{n\times m}$ denote the spaces of natural numbers, real numbers, positive real numbers, real n-vectors, and real $n\times m$-matrices, respectively, I_n denotes a $n\times n$ identity matrix; $\mathscr{L}\{\cdot\}$ is the Laplace transform and $\mathscr{L}^{-1}\{\cdot\}$ is the inverse Laplace transform; $\mathscr{L}_\infty$ denotes the space of bounded signals, $\mathscr{L}_2[t_0,\infty)$ denotes the space of square integrable functions, $\lambda_{\min}(A)$ and $\lambda_{\max}(A)$ corresponds to the minimum and maximum eigenvalues of A; $\min(\cdot)$ denote the minimum operator; $\mathrm{adj}(A)$ denote the adjoint matrix, $\det(A)$ is the matrix determinant, the norms $\|A\|=\sqrt{\lambda_{\max}(A^\top A)}$ and $\|x\|$ stand for the spectral and vector Euclidean norms, respectively; $\otimes$ is the symmetric Kronecker product, $\mathrm{vech}(\cdot)$ is the half-vectorization operator; where $x\in\mathbb{R}^n,\ A\in\mathbb{R}^{n\times n}$ and $n,m\in\mathbb{N}$.

2. Problem formulation

Consider the following linear time invariant continuous-time system [12],

$$\begin{aligned}\dot{x}(t) &= Ax(t)+Bu(t), x(t_0)=x_0\\ y(t) &= Cx(t)\end{aligned} \tag{1}$$

where $x(t) \in \mathbb{R}^n$ is the state, $u(t) \in \mathbb{R}^m$ is the control input, $y \in \mathbb{R}^p$ is the output, $A \in \mathbb{R}^{n\times n}, B \in \mathbb{R}^{n\times m}$, and $C \in \mathbb{R}^{p\times n}$ denote the plant, input, and output matrices respectively that will be considered unknown. It is assumed that the pair (A, B) is controllable and the pair (A, C) is observable.

The main goal is to find a stabilizing controller $u(t)$ that minimizes the next value function,

$$\begin{aligned} V(x(t)) &= \int_t^\infty \rho(y(\tau), u(\tau))d\tau \\ &= \int_t^\infty \left(y^\top(\tau)S_y y(\tau) + u^\top(\tau)Ru(\tau)\right)d\tau \end{aligned} \tag{2}$$

where $S_y > 0 \in \mathbb{R}^{p\times p}$ and $R > 0 \in \mathbb{R}^{m\times m}$ are predefined symmetric and positive definite weight matrices of the utility function $\rho(y(t), u(t))$ which are assumed to be diagonal. Notice that $y^\top(t)S_y y(t) = x^\top(t)Sx(t)$ with $S = C^\top S_y C \geqslant 0 \in \mathbb{R}^{n\times n}$.

The Hamiltonian associated with (1) and (2) is

$$\begin{aligned} H(x(t), u(t), \nabla V(t)) = \quad & \nabla V(t)\dot{x}(t) + x^\top(t)Sx(t) \\ & + u^\top(t)Ru(t). \end{aligned} \tag{3}$$

where $\nabla = \frac{\partial}{\partial x}$ is the gradient with respect to x. The optimal value function $V^*(x(t))$ is formulated as

$$V^*(x(t)) = \min_{u\in U} \int_t^\infty \left(x^\top(\tau)Sx(\tau) + u^\top(\tau)Ru(\tau)\right)d\tau. \tag{4}$$

The application of the Bellman optimality principle in (3) gives the following Hamilton–Jacobi-Bellman (HJB) equation

$$\min_{u\in U}\{H(x(t), u, \nabla V^*(t))\} = 0. \tag{5}$$

Since the system (1) is linear, then the optimal value function can be designed as a quadratic function in terms of the state, that is, $V^*(x(t)) : \mathbb{R}^n \to \mathbb{R}$,

$$V^*(x(t)) = x^\top(t)Px(t), \forall x, \tag{6}$$

where $P = P^\top \in \mathbb{R}^{n\times n}$ is a positive definite kernel matrix which is solution of the next Algebraic Riccati equation (ARE)

$$A^\top P + PA + S - PBR^{-1}B^\top P = 0. \tag{7}$$

The optimal control solution is obtained by employing the stationary condition $\frac{\partial H(\cdot)}{\partial u} = 0$, which results in

$$\begin{aligned} u^*(t) &= \arg\min_{u\in U} H(x(t), u, \nabla V^*(t)) \\ &= -Kx(t) = -R^{-1}B^\top Px(t). \end{aligned} \tag{8}$$

where $K \in \mathbb{R}^{m\times n}$ is the optimal control gain matrix. The optimal value function (6) can be equivalently expressed as the following Q-function $Q(x, u) : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}$ [36],

$$\begin{aligned} Q(x, u) &:= V^*(x) + H(x, u, \nabla V^*) \\ &= x^\top Px + P(Ax + Bu) + (Ax + Bu)^\top P \\ &\quad + x^\top Sx + u^\top Ru, \forall x, u. \end{aligned} \tag{9}$$

In this formulation the dependence of time is omitted for sake of simplicity. The Q-function (9) can be written as a quadratic function [39] in terms of the state x and control u as

$$\begin{aligned} Q(x, u) &= \begin{bmatrix} x \\ u \end{bmatrix}^\top \begin{bmatrix} P + A^\top P + PA + S & PB \\ B^\top P & R \end{bmatrix} \begin{bmatrix} x \\ u \end{bmatrix} \\ &= \begin{bmatrix} x \\ u \end{bmatrix}^\top \begin{bmatrix} Q_{xx} & Q_{xu} \\ Q_{xu}^\top & Q_{uu} \end{bmatrix} \begin{bmatrix} x \\ u \end{bmatrix} \end{aligned} \tag{10}$$

where $\begin{bmatrix} Q_{xx} & Q_{xu} \\ Q_{xu}^\top & Q_{uu} \end{bmatrix} \in \mathbb{R}^{(n+m)\times(n+m)}$. The stationary condition $\frac{\partial Q(\cdot)}{\partial u} = 0$ is employed to derive the optimal control as follows

$$\begin{aligned} u^*(t) &= \arg\min_{u\in U} Q(x, u) \\ &= \tfrac{\partial Q(x,u)}{\partial u} = -Q_{uu}^{-1}Q_{xu}^\top x(t). \end{aligned} \tag{11}$$

The optimal Q-function $Q^*(x, u^*)$ is equivalent to the optimal value function (6) because the Hamiltonian is equivalent to zero under the optimal control (11), that is, it satisfies the ARE (7).

The above solutions need complete knowledge of the system dynamics (1) and full state feedback which is assumed to be not available in this paper. Besides, the above statement is common in most real applications. To overcome the aforementioned issues, a state parametrization in terms of a Luenberger observer [12] is employed with a Q-learning update rule.

The following formulations are required before starting the observer design. The utility function in (2) can be factored as

$$\begin{aligned}\rho(y(t),u(t)) = &\begin{bmatrix}y(t)\\u(t)\end{bmatrix}^{\top}\underbrace{\begin{bmatrix}S_y & 0\\0 & R\end{bmatrix}}_{\mathscr{S}}\begin{bmatrix}y(t)\\u(t)\end{bmatrix}\\ = &\begin{bmatrix}y(t)\\u(t)\end{bmatrix}^{\top}M^{\top}M\begin{bmatrix}y(t)\\u(t)\end{bmatrix}\\ = &\begin{bmatrix}y(t)\\u(t)\end{bmatrix}^{\top}\underbrace{\begin{bmatrix}M_s^{\top}\\M_r^{\top}\end{bmatrix}\begin{bmatrix}M_s & M_r\end{bmatrix}}_{M^{\top}M}\begin{bmatrix}y(t)\\u(t)\end{bmatrix}\end{aligned} \tag{12}$$

where $M^{\top}M = \mathscr{U}\mathscr{D}\mathscr{U}^{\top} \in \mathbb{R}^{r\times r}$ denotes the spectral decomposition of matrix $\mathscr{S} \in \mathbb{R}^{r\times r}$, here $\mathscr{U} \in \mathbb{R}^{r\times r}$ is the matrix of eigenvectors, and $\mathscr{D} = \text{diag}\{\lambda_i\} \in \mathbb{R}^{r\times r}$ is a diagonal matrix whose main diagonal consists of the eigenvalues of matrix $\mathscr{S}$, here $r = p + m$. So $M = \mathscr{D}^{1/2}\mathscr{U}^{\top}$ which can also be factorized by the matrices $M_s \in \mathbb{R}^{r\times p}$ and $M_r \in \mathbb{R}^{r\times m}$. The control input $u(t)$ is finite and bounded due to the design of R, that is, $u(t) \in \mathscr{L}_2[t_0,\infty)$. Here $\mathscr{L}_2[t_0,\infty)$ is the space over which the integral is minimized [40]. Then, the optimal value function (4) can be expressed as

$$V^*(x(t)) = \min_{u\in\mathscr{L}_2[t_0,\infty)} \|M_sCx(t) + M_ru(t)\|^2. \tag{13}$$

So the LQR problem can be formulated as an optimization problem of the form

$$\begin{aligned}\min_{u\in\mathscr{L}_2[t_0,\infty)} \quad & \|M_sCx(t) + M_ru(t)\|^2\\ \text{s.t.} \quad & \dot{x}(t) = Ax(t) + Bu(t), x(t_0) = x_0.\end{aligned} \tag{14}$$

The above notation facilitates the state observer parametrization which is discussed in the next section.

3. State observer parametrization

The main objective is to define a state parametrization of the state $x(t)$ in terms of the control input $u(t)$ and the utility function factorization (12). First, let define a new output in terms of the utility function factorization

$$l(t) := M_sCx(t) + M_ru(t). \tag{15}$$

where $l(t) \in \mathbb{R}^r$. Notice that $\dim(y(t)) = p$ and $\dim(l(t)) = r$ where $r > p$. However, the observability of the pair (A, C) implies the observability of the pair (A, M_sC), that is, the rank of the observability matrices $\mathscr{O}(A, C) = \mathscr{O}(A, M_sC) = n$ and therefore, the following Luenberger Observer can be designed

$$\begin{aligned}&\dot{\widehat{x}}(t) = A\widehat{x}(t) + Bu(t) + L\Big(l(t) - \widehat{l}(t)\Big)\\ &= (A - LM_sC)\widehat{x}(t) + Bu(t) + LM_sy(t).\end{aligned} \tag{16}$$

where $\widehat{x}(t) \in \mathbb{R}^n$ is the estimate of the state $x(t)$, $\widehat{l}(t) = M_sC\widehat{x}(t) + M_ru(t)$ is the estimate of output $l(t)$, and $L \in \mathbb{R}^{n\times r}$ is the observer gain. The following Lemma establishes the convergence of $\widehat{x}(t)$ to $x(t)$ as $t \to \infty$.

Lemma 1. [40] Let the pair (A, M_sC) of system (1) and (15) be observable. If $u(t), l(t)$ are $\mathscr{L}_2[t_0,\infty)$ functions, then $x \in \mathscr{L}_2[t_0,\infty)$. Moreover, $\widehat{x}(t) \to x(t)$ as $t \to \infty$.

Proof. First of all, since the pair (A, M_sC) is observable then it is possible to design the observer gain L such that $(A - LM_sC)$ is Hurwitz. In consequence, from (16) is easy to check that $\widehat{x}(t) \in \mathscr{L}_2[t_0,\infty)$ because $u(t)$ and $l(t)$ are $\mathscr{L}_2[t_0,\infty)$ functions. Let define the observer error $\widetilde{x}(t) = x(t) - \widehat{x}(t)$; then the following dynamic equation is satisfied

$$\dot{\widetilde{x}}(t) = (A - LM_sC)\widetilde{x}(t), \tag{17}$$

whose solution is

$$\widetilde{x}(t) = \exp^{\bar{A}\sigma}\widetilde{x}_0, \tag{18}$$

where $\overline{A} = A - LM_sC$ and $\sigma = t - t_0$. Since $\overline{A}$ is Hurwitz then $\widetilde{x}(t) \in \mathscr{L}_2[t_0, \infty)$. Therefore $x(t) = \widehat{x}(t) + \widetilde{x}(t) \in \mathscr{L}_2[t_0, \infty)$. The above solution exhibits that the observer error exponentially converges to zero for any initial condition $\widetilde{x}_0$, that is, $\widetilde{x}(t) \to 0$ when $t \to \infty$ and hence, $\widehat{x}(t) \to x(t)$. This completes the proof.

Remark 1. The factorization $l(t)$ is used for the state-observer (16) design and not to solve the optimization problem (14).

Remark 2. Traditional Luenberger observers use the output of the system $y(t)$ to estimate the state $x(t)$ which its design is independent of the optimal control design. In contrast, the proposed approach uses the output $l(t)$ to correlate the estimation and optimal control problems in terms of the factored utility function (15).

In the sequel of this section the state parametrization is designed. The solution of (16) is

$$\widehat{x}(t) = \exp^{\overline{A}\sigma}\widehat{x}_0 + \int_{t_0}^{t} \exp^{\overline{A}(\sigma-\tau)}\left(Bu(\tau) + \overline{L}y(\tau)\right)d\tau \tag{19}$$

where $\overline{L} = LM_s$. The above solution does not give a feasible way to write a parametrization of the observer state $\widehat{x}(t)$ in terms of the control input $u(t)$ and the output $l(t)$. In the frequency domain, (19) is rewritten as [38]

$$\begin{aligned}\widehat{X}(s) = \ & \left(sI - \overline{A}\right)^{-1}\widehat{x}_0 + \left(sI - \overline{A}\right)^{-1} \\ & \times \left(\sum_{i=1}^{m} B_i U_i(s) + \sum_{i=1}^{p} \overline{L}_i Y_i(s)\right)\end{aligned} \tag{20}$$

where $\widehat{X}(s) = \mathscr{L}\{\widehat{x}(t)\}, U(s) = \mathscr{L}\{u(t)\}$, and $Y(s) = \mathscr{L}\{y(t)\}$. B_i and $\overline{L}_i$ denote the columns of matrix B and $\overline{L}$, respectively. Let define $D(s)$ as the characteristic polynomial of the Luenberger observer as

$$\begin{aligned} D(s) = \ & \det\left(sI - \overline{A}\right) \\ = \ & s^n + \alpha_{n-1}s^{n-1} + \ldots + \alpha_1 s + \alpha_0, \end{aligned}$$

where $\alpha_i > 0$. Notice that we can write the summations as follows

$$\begin{aligned} \sum_{i=1}^{m}\left(sI - \overline{A}\right)^{-1} B_i U_i(s) := \ & \sum_{i=1}^{m} \frac{\operatorname{adj}\left(sI - \overline{A}\right)}{D(s)} B_i U_i(s) \\ \sum_{i=1}^{p}\left(sI - \overline{A}\right)^{-1} \overline{L}_i Y_i(s) := \ & \sum_{i=1}^{p} \frac{\operatorname{adj}\left(sI - \overline{A}\right)}{D(s)} \overline{L}_i Y_i(s). \end{aligned} \tag{21}$$

Consider the first summation of (21). Then it is possible to split the numerator as

$$\begin{aligned} \operatorname{adj}\left(sI - \overline{A}\right)B_i = & \begin{bmatrix} \beta_{n-1}^{i1}s^{n-1} + \ldots + \beta_1^{i1}s + \beta_0^{i1} \\ \beta_{n-1}^{i2}s^{n-1} + \ldots + \beta_1^{i2}s + \beta_0^{i2} \\ \vdots \\ \beta_{n-1}^{in}s^{n-1} + \ldots + \beta_1^{in}s + \beta_0^{in} \end{bmatrix} \\ = & \begin{bmatrix} \beta_0^{i1} & \beta_1^{i1} & \cdots & \beta_{n-1}^{i1} \\ \beta_0^{i2} & \beta_1^{i2} & \cdots & \beta_{n-1}^{i2} \\ \vdots & \vdots & \vdots & \vdots \\ \beta_0^{in} & \beta_1^{in} & \cdots & \beta_{n-1}^{in} \end{bmatrix} \begin{bmatrix} 1 \\ s \\ \vdots \\ s^{n-1} \end{bmatrix} \\ * \ \triangleq & W_u^i N(s) \end{aligned}$$

where $W_u^i \in \mathbb{R}^{n\times n}$ contains the coefficients of the numerator and $N(s) \in \mathbb{R}^n$ is a vector of powers of s. So

$$\sum_{i=1}^{m} W_u^i \frac{N(s)}{D(s)} U_i(s). \tag{22}$$

The term $N(s)/D(s)$ defines n filters applied to the control input $u_i(t)$. This vector of n filters of $u_i(t)$ can be obtained by the following linear system

$$\begin{aligned} \dot{\xi}_u^i(t) \ & = A_L\xi_u^i(t) + B_L u_i(t), \xi_u^i(t_0) = 0, \\ w_u^i(t) \ & = I_n\xi_u^i(t) \end{aligned} \tag{23}$$

where $\xi_u^i(t) \in \mathbb{R}^n$ is the new state vector and $w_u^i(t) \in \mathbb{R}^n$ is its output. The matrices of (23) are defined as

$$A_L = \begin{bmatrix} 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \dots & 1 \\ -\alpha_0 & -\alpha_1 & -\alpha_2 & \dots & -\alpha_n \end{bmatrix}, B_L = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix}. \tag{24}$$

Notice that $\det(sI - A_L) = D(s)$. Finally the first summation of (21) can be written in the time domain as

$$\mathscr{L}^{-1}\left\{ \sum_{i=1}^{m} \frac{\operatorname{adj}\left(sI - \overline{A}\right)}{D(s)} B_i U_i(s) \right\} = W_u \xi_u(t). \tag{25}$$

The matrices are defined by $W_u = \begin{bmatrix} W_u^1 & \cdots & W_u^m \end{bmatrix} \in \mathbb{R}^{n \times nm}$ and $\xi_u(t) = \begin{bmatrix} \left(\xi_u^i\right)^\top(t) & \cdots & \left(\xi_u^m\right)^\top(t) \end{bmatrix}^\top \in \mathbb{R}^{nm}$. The second summation of (21) can be equivalently written as

$$\mathscr{L}^{-1}\left\{ \sum_{i=1}^{p} \frac{\operatorname{adj}\left(sI - \overline{A}\right)}{D(s)} \overline{L}_i Y_i(s) \right\} = W_l \xi_l(t), \tag{26}$$

where $W_l = \begin{bmatrix} W_l^i & \cdots & W_l^p \end{bmatrix} \in \mathbb{R}^{n \times np}$. Each W_l^i contains the coefficients of $\operatorname{adj}\left(sI - \overline{A}\right)\overline{L}_i$ and $\xi_l^i(t)$ is calculated from the next linear system

$$\begin{aligned} \dot{\xi}_l^i(t) &= A_L \xi_l^i(t) + B_L y_i(t), \xi_l^i(t_0) = 0, \\ w_l^i(t) &= I_n \xi_l^i. \end{aligned} \tag{27}$$

So, $\xi_l(t) = \begin{bmatrix} \left(\xi_l^i\right)^\top(t) & \cdots & \left(\xi_l^p\right)^\top(t) \end{bmatrix}^\top \in \mathbb{R}^{np}$. Since $\overline{A}$ is Hurwitz, then the first term of the right side of (19) converges to zero such that we can ignore it or assume that $\widehat{x}_0 = 0$. The final state parametrization is

$$\widehat{x}(t) = W_u \xi_u(t) + W_l \xi_l(t). \tag{28}$$

The next theorem establishes the optimality of the state feedback controller under state observer measurements.

Theorem 1. There exists a unique optimal control $u = -Kx$ for the LQR problem which is independent of the initial condition x_0 and satisfies

$$\min_{u \in U} \int_{t_0}^{\infty} \rho(x(\tau), u(\tau)) d\tau = x_0^\top P x_0. \tag{29}$$

Proof. See Appendix A.

The state parametrization (28) still require knowledge of the parameters of the system dynamics. In the next section the Q-learning algorithm is designed in terms of the state-parametrization.

4. Q-function new parametrization

The optimal value function (6) in terms of the state parameterization (28) is [38]

$$\begin{aligned} V_\xi^*(t) &= \begin{bmatrix} \xi_u(t) \\ \xi_l(t) \end{bmatrix}^\top \begin{bmatrix} W_u^\top & W_l^\top \end{bmatrix}^\top P \begin{bmatrix} W_u & W_l \end{bmatrix} \begin{bmatrix} \xi_u(t) \\ \xi_l(t) \end{bmatrix} \\ &= \begin{bmatrix} \xi_u(t) \\ \xi_l(t) \end{bmatrix}^\top \begin{bmatrix} W_u^\top P W_u & W_u^\top P W_l \\ W_l^\top P W_u & W_l^\top P W_l \end{bmatrix} \begin{bmatrix} \xi_u(t) \\ \xi_l(t) \end{bmatrix} \\ &= \xi^\top(t) P_\xi \xi(t), \end{aligned} \tag{30}$$

where $\xi(t) = \begin{bmatrix} \xi_u^\top(t) & \xi_l^\top(t) \end{bmatrix}^\top \in \mathbb{R}^j$, with $j = mn + np$, and

$$P_\xi = \begin{bmatrix} W_u^\top P W_u & W_u^\top P W_l \\ W_l^\top P W_u & W_l^\top P W_l \end{bmatrix} \in \mathbb{R}^{j \times j}.$$

Notice that (6) and (30) are equivalent. However P_ξ is not a kernel matrix because is a positive semi-definite matrix with rank n which depends on the real kernel matrix P. The Hamiltonian associated to (30) and (28) is

$$\begin{aligned} H_\xi(t) = \ & \frac{\partial V_\xi(t)}{\partial \xi_u}\dot{\xi}_u(t) + \frac{\partial V_\xi(t)}{\partial \xi_l}\dot{\xi}_l(t) + y^\top(t)S_y y(t) \\ & + u^\top(t)Ru(t). \end{aligned}$$

Here the utility function is written in terms of the output $y(t)$ and not the state $x(t)$. The HJB equation is written as

$$\begin{aligned} H_\xi(t) = \ & 2\left[\left[\xi_u^\top(t)W_u^\top PW_u + \xi_l^\top(t)W_l^\top PW_u\right]\{\mathscr{A}_m\xi_u(t) \right. \\ & + \mathscr{B}_m u(t)\} + \left[\xi_l^\top(t)W_l^\top PW_l + \xi_u^\top(t)W_u^\top PW_l\right] \\ & \left. \times\left\{\mathscr{A}_p\xi_l(t) + \mathscr{B}_p y(t)\right\}\right] + y^\top(t)S_y y(t) \\ & + u^\top(t)Ru(t) \end{aligned} \tag{31}$$

where $\mathscr{A}_m \in \mathbb{R}^{nm\times nm}, \mathscr{B}_m \in \mathbb{R}^{nm\times m}, \mathscr{A}_p \in \mathbb{R}^{np\times np}$, and $\mathscr{B}_p \in \mathbb{R}^{np\times p}$. Each matrix has the following form

$$\mathscr{A}_k = \begin{bmatrix} A_L & & 0 \\ & \ddots & \\ 0 & & A_L \end{bmatrix} \mathscr{B}_k = \begin{bmatrix} B_L & & 0 \\ & \ddots & \\ 0 & & B_L \end{bmatrix}, k = m, p.$$

Applying the stationary condition $\frac{\partial H_\xi}{\partial u} = 0$ gives the following optimal controller

$$\begin{aligned} u^*(t) \ & = \arg\min_{u\in U} H_\xi(t) \\ & = -R^{-1}\mathscr{B}_m^\top W_u^\top P(W_u\xi_u(t) + W_l\xi_l(t)). \end{aligned} \tag{32}$$

Here $W_u\mathscr{B}_m = B$ and (28) is satisfied. Hence the optimal control (32) is equivalent to (8).

The new Q-function can be expressed in terms of (30) and (31) as follows

$$\begin{aligned} Q_\xi(t) := \ & V_\xi^*(t) + H_\xi(t) \\ = \ & \begin{bmatrix} \xi_u(t) \\ \xi_l(t) \\ y(t) \\ u(t) \end{bmatrix}^\top \begin{bmatrix} Q_{\xi_u\xi_u} & Q_{\xi_u\xi_l} & Q_{\xi_u y} & Q_{\xi_u u} \\ Q_{\xi_u\xi_l}^\top & Q_{\xi_l\xi_l} & Q_{\xi_l y} & Q_{\xi_l u} \\ Q_{\xi_u y}^\top & Q_{\xi_l y}^\top & Q_{yy} & 0_{p\times m} \\ Q_{\xi_u u}^\top & Q_{\xi_l u}^\top & 0_{m\times p} & Q_{uu} \end{bmatrix} \begin{bmatrix} \xi_u(t) \\ \xi_l(t) \\ y(t) \\ u(t) \end{bmatrix} \\ = \ & z^\top(t)\mathscr{H}z(t) \end{aligned} \tag{33}$$

where $z(t) = \left[\xi_u^\top(t), \xi_l^\top(t), y^\top(t), u^\top(t)\right]^\top \in \mathbb{R}^q$ and $\mathscr{H} \in \mathbb{R}^{q\times q}$ with $q = j + p + m$ and

$$\begin{aligned} Q_{\xi_u\xi_u} = \ & 2W_u^\top PW_u\mathscr{A}_m + W_u^\top PW_u \in \mathbb{R}^{mn\times mn} \\ Q_{\xi_u\xi_l} = \ & \mathscr{A}_m^\top W_u^\top PW_l + W_u^\top PW_l\left(\mathscr{A}_p + I_p\right) \in \mathbb{R}^{mn\times np} \\ Q_{\xi_u y} = \ & W_u^\top PW_l\mathscr{B}_p \in \mathbb{R}^{mn\times p} \\ Q_{\xi_u u} = \ & W_u^\top PW_u\mathscr{B}_m \in \mathbb{R}^{mn\times m} \\ Q_{\xi_l\xi_l} = \ & 2W_l^\top PW_l\mathscr{A}_p + W_l^\top PW_l \in \mathbb{R}^{np\times np} \\ Q_{\xi_l y} = \ & W_l^\top PW_l\mathscr{B}_p \in \mathbb{R}^{np\times p} \\ Q_{\xi_l u} = \ & W_l^\top PW_u\mathscr{B}_m \in \mathbb{R}^{np\times m} \\ Q_{yy} = \ & S_y \in \mathbb{R}^{p\times p} \\ Q_{uu} = \ & R \in \mathbb{R}^{m\times m}. \end{aligned}$$

The stationary condition $\frac{\partial Q_\xi(t)}{\partial u} = 0$ is employed to derive the optimal control as follows

$$\begin{aligned} u^*(t) \ & = \arg\min_{u\in U} Q_\xi(t) \\ & = -Q_{uu}^{-1}\left(Q_{\xi_u u}^\top\xi_u(t) + Q_{\xi_l u}^\top\xi_l(t)\right) \\ & = -R^{-1}\mathscr{B}_m^\top W_u^\top P(W_u\xi_u(t) + W_l\xi_l(t)) \\ & = -R^{-1}B^\top P\widehat{x}(t). \end{aligned} \tag{34}$$

Notice that the state-control pair (x,u) of the Q-function (10) is expanded into a tuple (ξ_u,ξ_l,y,u). This tuple gives a large number of basis functions which are helpful for the Q-function identification. However, the number of parameters and computational effort are increased.

5. Q-learning formulation

The Q-function (33) can be expressed as

$$Q_{\bar{\xi}}(t)=z^{\top}(t)\mathscr{H}z(t)=\text{vech}(\mathscr{H})^{\top}(z(t)\otimes z(t)) \tag{35}$$

where $\Theta:=\text{vech}(\mathscr{H})\in\mathbb{R}^{\frac{1}{2}q(q+1)}$ is the half-vectorization of the matrix $\mathscr{H}$ where the off-diagonal elements are taken as $2\mathscr{H}_{ij}$.

Since Θ is unknown, then the following approximator is used

$$\widehat{Q}_{\bar{\xi}}(t)=\widehat{\Theta}^{\top}(z(t)\otimes z(t)) \tag{36}$$

where $\widehat{\Theta}\in\mathbb{R}^{\frac{1}{2}q(q+1)}$ are the estimates of Θ.

The optimal value function (30) can be written in the integral reinforcement learning (IRL) form [41,42,37] as

$$V^{*}_{\bar{\xi}}(t)=V^{*}_{\bar{\xi}}(t-T)-\int_{t-T}^{t}\rho(y(\tau),u(\tau))d\tau \tag{37}$$

where $T\in\mathbb{R}_{+}$ is a small fixed time. The following temporal difference error $\delta(t)$ is defined in terms of the approximator (36) as

$$\begin{aligned}\delta(t):=&\ \widehat{Q}_{\bar{\xi}}(t-T)-\widehat{Q}_{\bar{\xi}}(t)-\textstyle\int_{t-T}^{t}\rho(y(\tau),u(\tau))d\tau\\ =&\ \widehat{\Theta}^{\top}\Phi(t)-R_y(t).\end{aligned} \tag{38}$$

where $\Phi(t)=z(t-T)\otimes z(t-T)-z(t)\otimes z(t)$ and

$$R_y(t)=\int_{t-T}^{t}\rho(y(\tau),u(\tau))d\tau\in\mathbb{R}.$$

The value iteration algorithm is designed to minimize the following cost index [25]

$$E(t)=\frac{1}{2}\delta^{2}(t). \tag{39}$$

A gradient descent update rule[43] is used as

$$\begin{aligned}\dot{\widehat{\Theta}}(t)=&\ -\alpha\tfrac{\partial E(t)}{\partial\widehat{\Theta}}\\ =&\ -\alpha\Phi(t)\Big(\Phi^{\top}(t)\widehat{\Theta}(t)-R_y(t)\Big)\end{aligned} \tag{40}$$

where $\alpha\in\mathbb{R}_{+}$ is the learning rate. The update rule (40) can be equivalently written as

$$\dot{\widetilde{\Theta}}(t)=-\alpha\Phi(t)\Phi^{\top}(t)\widetilde{\Theta}(t) \tag{41}$$

where $\widetilde{\Theta}(t)=\widehat{\Theta}(t)-\Theta$ denote the parametric error.

Remark 3. The time T determines the step size of the integral. Hence a large enough T must be selected to obtain a fast convergence of the Q-learning algorithm. On the other hand, a large learning rate α could destabilize the closed-loop performance for a large time T. Conversely, a small learning rate α exhibits slow learning and, in most cases, will not achieve parameter convergence even the PE condition is satisfied. Therefore, an adequate selection of α and T is paramount to avoid instability and slow learning performance.

Parameter convergence is obtained if the following persistence of excitation (PE) condition [43] on the matrix $\Phi(t)$ is fulfilled.

Definition 1. [16] A vector $\Phi(t):\mathbb{R}^{q}\to\mathbb{R}^{\frac{1}{2}q(q+1)}$ is persistently exciting (PE) if there exists $\beta_1,\beta_2,T>0$ such that for all $t\geqslant 0$ the next relationship is fulfilled

$$\beta_1 I_q\leqslant\int_{t-T}^{t}\Phi(\tau)\Phi^{\top}(\tau)d\tau\leqslant\beta_2 I_q. \tag{42}$$

The above definition requires that the $q\times q$ matrix $\Phi(t)\Phi^{\top}(t)$ integrated over the interval $[t-T:t]$ be nonsingular.

Theorem 2. Let $\Phi(t)$ be PE. The error dynamics (41) converges exponentially to zero as $t \to \infty$ and hence, the estimates $\widehat{\Theta}(t) \to \Theta$.

Proof. See Appendix A.

The Q-learning formulation (40) avoids knowledge of the linear system dynamics (1) and the parametrization matrices W_u and W_l of (28). In this case, only measurements of the output y, the input u, and the Luenberger states ξ_u and ξ_l are required to compute the optimal control policy. Here the regressor vector $\Phi(t)$ contains more signals that offer a rich enough excitement for parameter convergence if the control input u fulfils the PE condition (42). However, the measurements of the states x are avoided by increasing the dimensionality of the Q-function parametrization and hence, the computational complexity is increased.

6. Simulation studies

In this section the performance of the proposed approach is verified by: (1) analysing the performance of the state parametrization under the new output, (2) showing that the estimates converge to the optimal kernel solution of the LQR problem, (3) demonstrate that the state parametrization spans the number of basis functions for the value function approximation such that it improves the excitement of the regressor matrix $\Phi(t)$.

The F-16 short period dynamics proposed in [24] is used to evaluate the proposed approach. The system dynamics is written as in (1) with

$$A = \begin{bmatrix} -1.01887 & 0.90506 & -0.00215 \\ 0.82225 & -1.07741 & -0.17555 \\ 0 & 0 & -20.2 \end{bmatrix} B = \begin{bmatrix} 0 \\ 0 \\ 20.2 \end{bmatrix}$$
$$C = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}.$$

Here, $x = [\eta, q, \delta_e]^\top$, where η is the angle of attack, q is the pitch rate and δ_e is the elevator deflection angles, and $u = \delta_{ec}$ is the elevator command. Is easy to check that the pair (A, B) is controllable and the pair (A, C) is observable. The weight matrices of the utility function were proposed as $S_y = I_2$ and $R = 1$. Under these conditions $M_s = C$ and M_r could be chosen as $M_r = [0, 1]^\top$ or $M_r = [1, 0]^\top$. M_r does not affect the observer design. The desired eigenvalues of the state observer were located at $\lambda_i = -2, -3, -20$. The observer gain and the exact solution of the LQR problem (control gain and kernel solution) were computed off-line. The obtained results were

$$L = \begin{bmatrix} 0.9812 & 0.8099 & -0.2656 \\ 0.8933 & 1.7225 & -19.5928 \end{bmatrix}^\top$$
$$K = \begin{bmatrix} -0.1951 & -0.2368 & 0.0021 \end{bmatrix}$$
$$P = \begin{bmatrix} 1.3581 & 1.0979 & -0.0097 \\ 1.0979 & 1.3603 & -0.0117 \\ -0.0097 & -0.0117 & 0.0001 \end{bmatrix}.$$

The exact weight matrices of the state parametrization were

$$W_u = \begin{bmatrix} -0.1634 & -0.0434 & 0 \\ -7.093 & -3.5461 & 0 \\ 113.1181 & 96.96 & 20.2 \end{bmatrix}$$
$$W_l = \begin{bmatrix} 59.0305 & 22.5775 & 0.9812 \\ 33.0149 & 18.0398 & 0.8099 \\ 31.2209 & 14.8550 & -0.2656 \\ 54.0912 & 20.6079 & 0.8933 \\ 76.6532 & 41.6907 & 1.7225 \\ -41.3326 & -60.0591 & -19.5928 \end{bmatrix}.$$

First the performance of the state parametrization was tested. The aircraft was controlled in open-loop with a control input of $u(t) = 0.1 \sin(\pi t)$. Fig. 1 exhibits the state observer parametrization (SP) results and the equivalence with the Luenberger Observer (LO) results. As it is observed, the SP is equivalent to the LO dynamics and hence, the states estimates converge to their real state values.

The Q-function $Q(x, u)$ and $Q_\xi(t)$ were compared with the LQR solution. Whilst $Q(x(t))$ had 9 parameters, Q_ξ had 72 parameters (Q_{uu} and Q_{yy} were not estimated because they were known values). A PE signal composed of a sum of exponen-

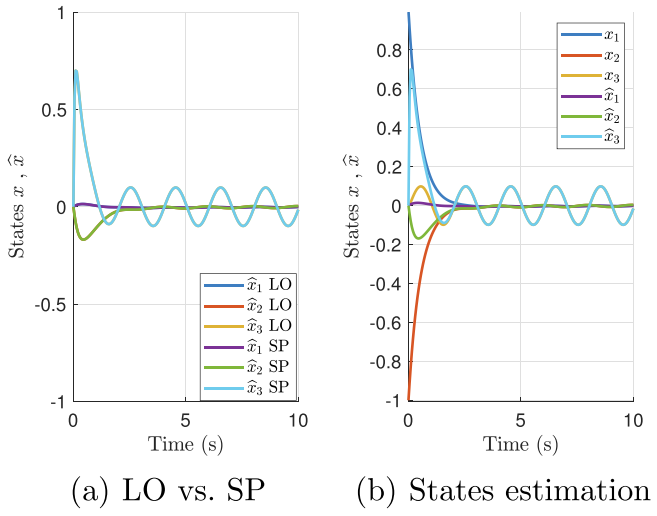

Fig. 1. State observer parametrization results.

tial sinusoidal functions with different time frequencies was used to excite the modes of the Aircraft dynamics in the first 15 s of the simulation time. The IRL used a fixed time $T = 0.05$ s. The learning rate was manually tuned until the best performance was achieved. The best learning rate was $\alpha = 30$. Fig. 2 shows the time evolution and stabilization of the aircraft states.

Notice that the trajectories of the unforced aircraft dynamics ($u(t) = 0$) are asymptotically stable. The estimated optimal control laws of the Q-learning with full state feedback (QL FS) and the Q-learning with state parametrization (QL SP) were

$$\widehat{u}_{FS}(t) = -[-0.1959 \quad -0.2363 \quad 0.0020]x(t),$$

$$\widehat{u}_{SP}(t) = -\begin{bmatrix} 1.9369 \\ 1.0424 \\ 0.0408 \end{bmatrix}^{\top} \xi_u(t) + \begin{bmatrix} 19.3061 \\ 8.6574 \\ 0.3842 \\ 28.7983 \\ 14.0124 \\ 0.6217 \end{bmatrix}^{\top} \xi_l(t).$$

The performance of each optimal controller under the PE signal is observed in Fig. 3. All controllers exhibit the same performance and converge to the optimal control policy.

Fig. 4 shows the minimization of the temporal difference error (which implicitly shows parameter estimates convergence) in the first ten seconds of simulation time.

The Euclidean norm of the parameter estimates error $\|\widetilde{\Theta}\|$ was used as performance metric. The results were $\|\widetilde{\Theta}_{FS}\| = 9.951 \times 10^{-4}$ and $\|\widetilde{\Theta}_{SP}\| = 4.49 \times 10^{-2}$. It is clear that the parametric error was increased by incorporating new parameters. However, the final near optimal controller is closed to the LQR solution as it is shown Fig. 3. The performance of each controller without the PE signal is shown in Fig. 5.

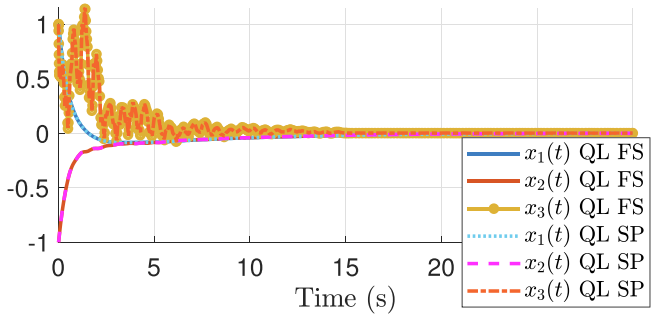

Fig. 2. Time evolution of the states.

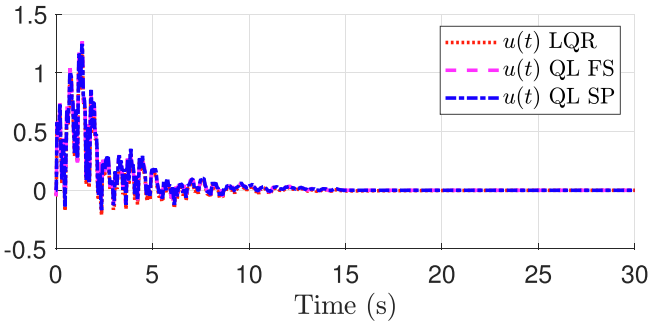

Fig. 3. Control input comparison.

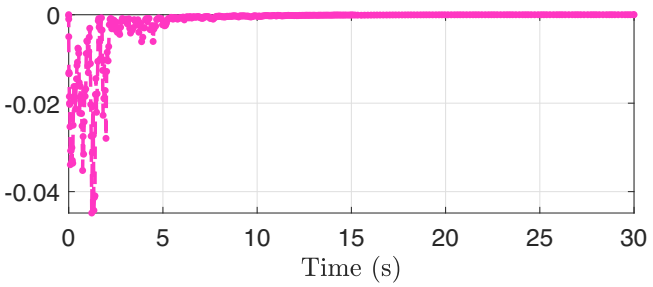

Fig. 4. Temporal difference error minimization.

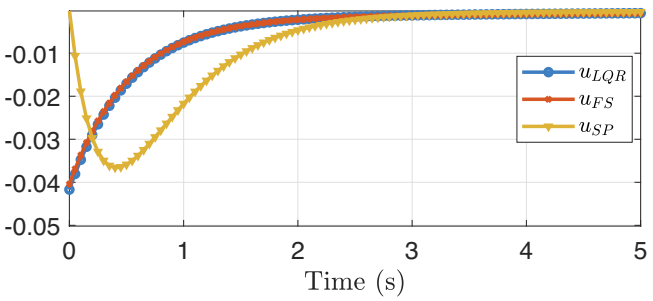

Fig. 5. Optimal controllers comparison.

Notice that the optimal control u_{SP} does not have knowledge of the initial condition of the aircraft dynamics. The states trajectories stabilize at the origin almost at the same time of the other optimal controllers. Robustness of optimal control law was achieved by adding more basis functions and estimates at the final optimal control law. This fact provides adaptability of the estimates to changes at the black box system.

7. Conclusions

In this paper a Q-learning algorithm for black box linear systems is proposed. A state parametrization composed by a Luenberger observer and an utility function factorization is used to estimate the internal states of the linear system and to define a new Q-function parametrization in terms of the input, output, and the states of the parametrization. The IRL and gradient descent formulations are used to estimate on-line the parameters of the optimal Q-function. Stability and uniqueness of the optimal solution are analysed using Lyapunov stability theory. Simulations are carried out to verify the approach which shows: i) a stable states parametrization, ii) near optimal solution of the Q-learning algorithm under the assumption of full state feedback and the state parametrization, iii) adding more basis functions and estimates to the Q-function increases the parametric error norm and the robustness of the final optimal controller.

Future work will analyse deep learning architectures in the control context of adaptive dynamic programming and reinforcement learning.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgement

This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors.

Appendix A. Proofs

.

Proof. Proof of Theorem 1. Two cases are considered: (i) the feedback control $u(t) = -Kx(t)$ in (8) and (ii) the feedback control $u(t) = -K\widehat{x}(t)$. In both cases the gain K is the same.

Consider case (i). Substituting the feedback control (8) in the optimization problem (14) gives

$$\begin{aligned} \dot{x}(t) &= (A - BK)x(t) := A_K x(t) \\ l(t) &= (M_s C - M_r K)x(t) := M_K x(t) \end{aligned} \tag{43}$$

and the ARE (7) is reduced to a Lyapunov equation of the form

$$A_K^\top P + P A_K + M_K^\top M_K = 0. \tag{44}$$

Notice that with this simplification it is evident that P is the observability Gramian of (A_K, M_K) which satisfies

$$\int_{t_0}^{\infty} \|l(\tau)\|^2 d\tau = x_0^\top P x_0. \tag{45}$$

So the control $u = -Kx(t)$ is called the optimal control law for the optimization problem (14). For case (ii), the feedback controller is rewritten as $u(t) = -K\widehat{x}(t) = -K(x(t) + \widetilde{x}(t)) := -Kx(t) + v(t)$. With this feedback controller the system (43) is rewritten as

$$\begin{bmatrix} \dot{x}(t) \\ l(t) \end{bmatrix} = \begin{bmatrix} A_K & B \\ M_K & M_r \end{bmatrix} \begin{bmatrix} x(t) \\ v(t) \end{bmatrix}. \tag{46}$$

From Lemma 1 we have that the signals $u(t), x(t), \widehat{x}(t), \widetilde{x}(t)$ and $l(t)$ are $\mathscr{L}_2[t_0, \infty)$ functions. So $v(t) \in \mathscr{L}_2[t_0, \infty)$. Moreover, $x(\infty) = 0$ because A_K is Hurwitz. Therefore, the next equivalence hold

$$\min_{u \in \mathscr{L}_2[t_0,\infty)} \|l(t)\|^2 = \min_{v \in \mathscr{L}_2[t_0,\infty)} \|l(t)\|^2. \tag{47}$$

By differentiating $x^\top(t)Px(t)$ with respect to t gives

$$\begin{aligned} & \tfrac{d}{dt}(x^\top(t)Px(t)) = 2x^\top(t)P(A_K x(t) + Bv(t)) \\ = \; & -x^\top(t)M_k^\top M_k x(t) + 2x^\top(t)PBv(t) \\ = \; & -(M_K x(t) + M_r v(t))^\top (M_K x(t) + M_r v(t)) \\ & +2x^\top(t)M_K^\top M_r v(t) + v^\top(t)M_r^\top M_r v(t) \\ & +2x^\top(t)PBv(t) \\ = \; & -\|l(t)\|^2 + \lambda_{\max}(R)\|v(t)\|^2. \end{aligned} \tag{48}$$

Integrating both sides of (48) in an interval $[t_0, \infty)$ gives

$$\begin{aligned} -x_0^\top P x_0 = \; & -\int_{t_0}^{\infty} \left(\|l(\tau)\|^2 - \lambda_{\max}(R)\|v(\tau)\|^2 \right) d\tau \\ \int_{t_0}^{\infty} \|l(\tau)\|^2 d\tau = \; & x_0^\top P x_0 + \lambda_{\max}(R) \int_{t_0}^{\infty} \|v(\tau)\|^2 d\tau. \end{aligned} \tag{49}$$

Notice that the unique optimal solution is when $v(t) = 0$, that is, $u(t) = Kx(t)$ which eventually happens since $\widetilde{x}(t) \to 0$ as $t \to \infty$ as it was stated in Lemma 1. Therefore the unique optimal control law is $u = -Kx(t)$ and the use of the state-observer does not affect the final closed-loop performance. This completes the proof.

Proof. Proof of Theorem 2. Consider the following Lyapunov function

$$\mathscr{V}(t) = \frac{1}{2}\widetilde{\Theta}^{\top}(t)\alpha^{-1}\widetilde{\Theta}(t).$$

The time-derivative of (50) along the trajectories of (41) is

$$\begin{aligned}\dot{\mathscr{V}}(t) = &\quad \widetilde{\Theta}(t)\alpha^{-1}\dot{\widetilde{\Theta}}(t)\\ = &\quad -\tfrac{1}{2}\widetilde{\Theta}^{\top}(t)\Phi(t)\Phi^{\top}(t)\widetilde{\Theta}(t).\end{aligned} \tag{51}$$

From (51), it is clear that $\widetilde{\Theta}(t)$ is an $\mathscr{L}_{\infty}$ function and $\mathscr{V}(t_0) \geqslant \mathscr{V}(t)$. On the other hand, boundedness of $\widetilde{\Theta}(t)$ implies boundedness of $\Phi(t)$, that is, $\Phi(t) \in \mathscr{L}_{\infty}$.

Let multiply (51) by one $\alpha\alpha^{-1} = 1$ as,

$$\begin{aligned}\dot{\mathscr{V}}(t) &= -\widetilde{\Theta}^{\top}(t)\Phi(t)\Phi^{\top}(t)\alpha\alpha^{-1}\widetilde{\Theta}(t)\\ &\leqslant -\gamma\mathscr{V}(t)\end{aligned} \tag{52}$$

where $\gamma = \alpha\lambda_{\min}\left(\Phi(t)\Phi^{\top}(t)\right)$. The solution of the above differential inequality is,

$$\mathscr{V}(t) \leqslant e^{-\gamma(t-t_0)}\mathscr{V}(t_0). \tag{53}$$

So

$$\begin{aligned}\tfrac{1}{\alpha}\|\widetilde{\Theta}(t)\|^2 \leqslant &\quad \widetilde{\Theta}^{\top}(t)\alpha^{-1}\widetilde{\Theta}(t) = \mathscr{V}(t)\\ \leqslant &\quad e^{-\gamma(t-t_0)}\widetilde{\Theta}^{\top}(t_0)\tfrac{1}{\alpha}\widetilde{\Theta}(t_0)\\ \leqslant &\quad \tfrac{1}{\alpha}e^{-\gamma(t-t_0)}\|\widetilde{\Theta}(t_0)\|^2.\end{aligned}$$

Hence, the parametric error of (41) converges exponentially to zero and satisfies

$$\|\widetilde{\Theta}(t)\| \leqslant e^{-\frac{1}{2}\gamma(t-t_0)}\|\widetilde{\Theta}(t_0)\| \tag{54}$$

This completes the proof.

References

[1] J. Chen, J. Su, J. Li, Self-coupling black box model of a dynamic system based on ann and its application, Math. Probl. Eng. (2020), https://doi.org/10.1155/2020/5724831.

[2] L. Ljung, Black-box models from input-output measurements, in: Imtc 2001. proceedings of the 18th ieee instrumentation and measurement technology conference. rediscovering measurement in the age of informatics (cat. no. 01ch 37188), Vol. 1, IEEE, 2001, pp. 138–146. doi:10.1109/IMTC.2001.928802..

[3] E. De la Rosa, W. Yu, H. Sossa, Fuzzy modeling from black-box data with deep learning techniques, International Symposium on Neural Networks, Springer (2017) 304–312, https://doi.org/10.1007/978-3-319-59072-1.

[4] Y.H. Kim, F.L. Lewis, Neural network output feedback control of robot manipulators, IEEE Trans. Robot. Autom. 15 (2) (1999) 301–309, https://doi.org/10.1109/70.760351.

[5] W. Yu, A. Perrusquía, Simplified stable admittance control using end-effector orientations, Int. J. Soc. Robot. 12 (5) (2020) 1061–1073, https://doi.org/10.1007/s12369-019-00579-y.

[6] Y. Chen, C. Hu, Y. Qin, M. Li, X. Song, Path planning and robust fuzzy output-feedback control for unmanned ground vehicles with obstacle avoidance, Proc. Inst. Mech. Eng., Part D: J. Autom. Eng. 235 (4) (2021) 933–944, https://doi.org/10.1177/0954407020978319.

[7] R. Cortois, G.B. De Deyn, The curse of the black box, Plant Soil 350 (1) (2012) 27–33, https://doi.org/10.1007/s11104-011-0963-z.

[8] A. Perrusquía, W. Yu, Human-in-the-loop control using euler angles, J. Intell. Robot. Syst. 97 (2) (2020) 271–285, https://doi.org/10.1007/s10846-019-01058-2.

[9] J.A. Flores-Campos, A. Perrusquía, L.H. Hernández-Gómez, N. González, A. Armenta-Molina, Constant speed control of slider-crank mechanisms: A joint-task space hybrid control approach, IEEE Access 9 (2021) 65676–65687, https://doi.org/10.1109/ACCESS.2021.3073364.

[10] D. Luviano, W. Yu, Continuous-time path planning for multi-agents with fuzzy reinforcement learning, J. Intell. Fuzzy Syst. 33 (1) (2017) 491–501, https://doi.org/10.3233/JIFS-161822.

[11] A. Perrusquía, J.A. Flores-Campos, W. Yu, Optimal sliding mode control for cutting tasks of quick-return mechanisms, ISA transactions doi:10.1016/j.isatra.2021.04.033..

[12] C.-T. Chen, B. Shafai, Linear system theory and design, vol. 3, Oxford University Press, New York, 1999.

[13] J.A. Moreno, Levant's arbitrary order differentiator with varying gain, IFAC-PapersOnLine 50 (1) (2017) 1705–1710, https://doi.org/10.1016/j.ifacol.2017.08.496.

[14] C. Edwards, S.K. Spurgeon, C.P. Tan, N. Patel, Sliding-mode observers, in: Mathematical methods for robust and nonlinear control, Springer, 2007, pp. 221–242..

[15] R. Sutton, A. Barto, Reinforcement Learning: An Introduction, Cambridge, MIT Press, MA, 1998.

[16] F.L. Lewis, D. Vrabie, V.L. Syrmos, Optimal control, John Wiley & Sons, 2012.

[17] F.L. Lewis, D. Vrabie, K.G. Vamvoudakis, Reinforcement learning and feedback control: Using natural decision methods to design optimal adaptive controllers, IEEE Control Syst. Mag. 32 (6) (2012) 76–105, https://doi.org/10.1109/MCS.2012.2214134.

[18] J. Zhang, Z. Wang, H. Zhang, Data-based optimal control of multiagent systems: A reinforcement learning design approach, IEEE Trans. Cybern. 49 (12) (2018) 4441–4449, https://doi.org/10.1109/TCYB.2018.2868715.

[19] B. Luo, H.-N. Wu, Huand-Tingwen, D. Liu, Reinforcement learning solution for HJB equation arising in constrained optimal control problem, Neural Networks 71 (2015) 150–158, https://doi.org/10.1016/j.neunet.2015.08.007.

[20] B. Kiumarsi, G.V. Kyriakos, H. Modares, F.L. Lewis, Optimal and autonomous control using reinforcement learning: A survey, IEEE Trans. Neural Networks Learn. Syst. 29 (6) (2018) 2042–2062, https://doi.org/10.1109/TNNLS.2017.2773458.

[21] H. Modares, F. Lewis, M.-B. Naghibi-Sistani, Integral reinforcement learning and experience replay for adaptive optimal control of partially-unknown constrained-input continuous-time systems, Automatica 50 (2014) 193–202, https://doi.org/10.1016/j.automatica.2013.09.043.
[22] A. Perrusquía, W. Yu, Identification and optimal control of nonlinear systems using recurrent neural networks and reinforcement learning: An overview, Neurocomputing 438 (2021) 145–154, https://doi.org/10.1016/j.neucom.2021.01.096.
[23] Y. Jiang, Z.-P. Jiang, Computational adaptive optimal control for continuous-time linear systems with completely unknown dynamics, Automatica 48 (10) (2012) 2699–2704, https://doi.org/10.1016/j.automatica.2012.06.096.
[24] J.-H. Kim, F. Lewis, Model-free H_∞ control design for unknown linear discrete-time systems via Q-learning with LMI, Automatica 46 (2010) 1320–1326, https://doi.org/10.1016/j.automatica.2010.05.002.
[25] K.G. Vamvoudakis, F.L. Lewis, Online actor–critic algorithm to solve the continuous-time infinite horizon optimal control problem, Automatica 46 (5) (2010) 878–888, https://doi.org/10.1016/j.automatica.2010.02.018.
[26] A. Perrusquía, W. Yu, Robust control under worst-case uncertainty for unknown nonlinear systems using modified reinforcement learning, Int. J. Robust Nonlinear Control 30 (7) (2020) 2920–2936, https://doi.org/10.1002/rnc.4911.
[27] B. Kiumarsi, F.L. Lewis, H. Modares, A. Karimpour, M.-B. Naghibi-Sistani, Reinforcement Q-learning for optimal tracking control of linear discrete-time systems with unknown dynamics, Automatica 50 (4) (2014) 1167–1175, https://doi.org/10.1016/j.automatica.2014.02.015.
[28] A. Perrusquía, W. Yu, A. Soria, Large space dimension reinforcement learning for robot position/force discrete control, in: 2019 6th International Conference on Control, Decision and Information Technologies (CoDIT), 2019, pp. 91–96, https://doi.org/10.1109/CoDIT.2019.8820575.
[29] H. Modares, F.L. Lewis, Linear quadratic tracking control of partially-unknown continuous-time systems using reinforcement learning, IEEE Trans. Autom. Control 59 (11) (2014) 3051–3056, https://doi.org/10.1109/TAC.2014.2317301.
[30] A. Perrusquía, W. Yu, X. Li, Nonlinear control using human behavior learning, Inf. Sci. 569 (2021) 358–375, https://doi.org/10.1016/j.ins.2021.03.043.
[31] A. Perrusquía, W. Yu, A. Soria, Position/force control of robot manipulators using reinforcement learning, Ind. Robot 46 (2) (2019) 267–280, https://doi.org/10.1108/IR-10-2018-0209.
[32] F. Lewis, S. Jagannathan, A. Yeşildirek, Neural Network control of robot manipulators and nonlinear systems, Taylor & Francis, 1999.
[33] A. Perrusquía, W. Yu, Discrete-time $\mathscr{H}_2$ neural control using reinforcement learning, IEEE Trans. Neural Networks Learn. Syst.https://doi.org/10.1109/TNNLS.2020.3026010..
[34] D. Vrabie, O. Pastravanu, M. Abu-Khalaf, F.L. Lewis, Adaptive optimal control for continuous-time linear systems based on policy iteration, Automatica 45 (2) (2009) 477–484, https://doi.org/10.1016/j.automatica.2008.08.017.
[35] A. Perrusquia, W. Yu, X. Li, Multi-agent reinforcement learning for redundant robot control in task-space, Int. J. Mach. Learn. Cybern. 12 (1) (2021) 231–241, https://doi.org/10.1007/s13042-020-01167-7.
[36] K.G. Vamvoudakis, Q-learning for continuous-time linear systems: A model-free infinite horizon optimal control approach, Syst. Control Lett. 100 (2017) 14–20, https://doi.org/10.1016/j.sysconle.2016.12.003.
[37] L.M. Zhu, H. Modares, G.O. Peen, F.L. Lewis, B. Yue, Adaptive suboptimal output-feedback control for linear systems using integral reinforcement learning, IEEE Trans. Control Syst. Technol. 23 (1) (2015) 264–273, https://doi.org/10.1109/TCST.2014.2322778.
[38] S.A.A. Rizvi, Z. Lin, Output feedback Q-learning control for the discrete-time linear quadratic regulator problem, IEEE Trans. Neural Networks Learn. Syst. 30 (5) (2019) 1523–1536, https://doi.org/10.1109/TNNLS.2018.2870075.
[39] T. Feng, J. Zhang, Y. Tong, H. Zhang, Q-learning algorithm in solving consensusability problem of discrete-time multi-agent systems, Automatica 128 (2021), https://doi.org/10.1016/j.automatica.2021.109576 109576.
[40] H. Khalil, Nonlinear systems, Prentice Hall, 2002.
[41] A. Perrusquía, W. Yu, Continuous-time reinforcement learning for robust control under worst-case uncertainty, Int. J. Syst. Sci. 52 (4) (2021) 770–784, https://doi.org/10.1080/00207721.2020.1839142.
[42] M. Palanisamy, H. Modares, F.L. Lewis, M. Aurangzeb, Continuous-time Q-learning for infinite-horizon discounted cost linear quadratic regulator problems, IEEE Trans. Cybern. 45 (2) (2015) 165–176, https://doi.org/10.1109/TCYB.2014.2322116.
[43] A. Perrusquía, W. Yu, Neural $\mathscr{H}_2$ control using continuous-time reinforcement learning, IEEE Trans. Cybern.https://doi.org/10.1109/TCYB.2020.3028988..