A LEARNING CONTROLLER DESIGN APPROACH FOR A 3-DOF HELICOPTER SYSTEM WITH ONLINE OPTIMAL CONTROL

Guilherme B. Sousa[1*], Janes V. R.Lima[1], Patrícia H. M.Rêgo[2], Alain G.Souza[3] and João V. FonsecaNeto[4]

[1]Postgraduate Program in Computer Engineering and Systems, State University of Maranhão -UEMA, São Luís, MA, Brazil
[2]Mathematics and Computing Department, State University of Maranhão - UEMA, São Luís,MA, Brazil
[3]Technological Institute of Aeronautics, São José dos Campos, SP, Brazil
[4]Department of Electrical Engineering, Federal University of Maranhão - UFMA, São Luís, MA, Brazil

ABSTRACT

This paper presents the design and investigation of performance of a 3-DOF Quanser helicopter system using a learning optimal control approach that is grounded on approximate dynamic programming paradigms, specifically action-dependent heuristic dynamic programming (ADHDP). This approach results in an algorithm that is embedded in the actor-critic reinforcement learning architecture, that characterizes this design as a model-free structure. The developed methodology aims at implementing an optimal controller that acts in real-time in the plant control, using only the input and output signals and states measured along the system trajectories. The feedback control design technique is capable of an online tuning of the controller parameters according to the plant dynamics, which is subject to the model uncertainties and external disturbances. The experimental results demonstrate the desired performance of the proposed controller implemented on the 3-DOF Quanser helicopter.

KEYWORDS

Action-Dependent Heuristic Dynamic Programming, Actor-Critic Reinforcement Learning, Real-Time Control, 3-DOF Helicopter.

1. INTRODUCTION

The safe, reliable and efficient control of the complex systems (in which there are aircrafts, automobiles, electric energy systems, etc.) is essential for our society. Such automatic decision and control systems are omnipresent in the modern engineering techniques and have an enormous impact in our lives. The intrinsic complexity of such systems shows the need of improvements of decision and control methods which provide a guaranteed performance and the satisfaction of the prescribed objectives [1].

The optimazation of sequential decisions or controls that are repeated throughout the process comes in various fields, and the optimal control theory provides methods to compute feedback control systems that supply an optimal performance. The dynamic programming is a useful technique to deal with optimal control problems, although it is usually sensitive to computing, once the execution costs are very high, caused by the "curse of dimensionality" [2] to execute it

Dhinaharan Nagamalai et al. (Eds) : CSE, AI&FL, CCNET, DBDM, CSIP, CICS - 2019
pp. 07-23, 2019. © CS & IT-CSCP 2019 DOI: 10.5121/csit.2019.90502

and to obtain better solutions. The controllers optimize the performance functions prescribed by the user and normally are designed offline when solving the Hamilton-Jacobi-Bellman (HJB) equations. That requires knowledge of a complete model of the system dynamics. Nevertheless, in many situations it is difficult to determine the precise dynamic model for practical systems.
In nature, most organisms act in an optimal way to maintain resources while achieving their objectives. Such principle, characterized by strong self-learning and adaptation abilities, substantiates the approximate/adaptive dynamic programming (ADP), proposed by Werbos [3, 4], which has shown to be efficient to determine real-time optimal control policies in solving Hamilton-Jacobi Bellman (HJB) design equations online, forward in time, and it becomes an important method of intelligent control for non-linear systems [1, 5, 6].

In the approximate dynamic programming context, incremental methods are usually used to solve the online learning problem of critical network parameters to approximate a value function. Among the proposed iterative algorithms to estimate such parameters, we highlight the recursive least squares (RLS) learning. The efficiency of RLS methods in incremental actor-critical learning is mainly due to its robustness to deal with time variations in regression parameters and fast convergence speed compared to stochastic gradient methods [7].

RLS learning is emphasized under the perspective of the research and the development of ADP based control systems. Actor critic structures based on RLS were proposed in [8] to improve the efficiency of the conventional heuristic adaptive critic methods. In [9] and [10] the authors explore RLS methods to solve learning problems concerning the actor-critic reinforcement. Pietquin and others [11] presented a recent advance in temporal difference (TD) methods such as Kalman temporal difference (KTD). In this scheme, a Kalman filter is embodied in the estimative of the approximation process of the value function using a state space representation. A prior development on KTD paradigms is presented in the work by Geist and others [12] for deterministic markovi an decision processes.

For many traditional iterative ADP algorithms, it is necessary to build a non-linear system model and, then, execute the ADP algorithms to derive an improved control policy. In terms of RLS learning to solve the discrete-time algebraic Riccati equation (DARE), also known as HJB-Riccati equation, in optimal control problems that are solved by the Heuristic Dynamic Programming (HDP) approach, the authors [13] developed methods and algorithms based on the RLS training for the online design of the discrete-time linear-quadratic regulator (DLQR).

In contrast to the HDP approach, the Q learning, proposed by [14, 15], is an ADP algorithm based on data, which has been called action-dependent heuristic dynamic programming (ADHDP) [16]. For the Q learning algorithms, the function Q is used rather than the cost function of the traditional iterative ADP algorithms. The function Q, also termed as action value function, depends on both the state x and the control action u, which means that it includes the information about the system and the cost function, making the obtaining process of control policies from the function Q easier than through the traditional functions of performance index. [17]. For such characteristics, algorithms based on Q learning are preferable to obtain the optimal control for systems with unknown dynamics exclusively from the observed data throughout the system trajectories [17].

This work presents the conception of an RLS-based ADHDP algorithm for the online optimal control problem solution. Such iterative learning algorithm is based on policy iteration principle, where the policy improvements are performed at every time step along the realization of state trajectory towards the optimal policy. This control strategy is directed to the operation of a plant in the form of a helicopter with three degrees of freedom (3-DOF helicopter) which aims to represent in a simple way the dynamic of a real helicopter with two counter-rotating propellers that exempt the necessity of a tail rotor. In order to compensate for the effects of disturbances and

variations in plant dynamics, the control system is designed so that the DLQR controller acts in real-time in plant control using only input and output signals measured along the system trajectory.

This work is organized as follows: Section 2 provides the system of equations of the 3-DOF Quanser helicopter system as well as its representation in state space, needed for the construction of the reference system to the proposed model. Section 3 provides, briefly, the fundamentals to assemble the optimal control design framework, which substantiates in the Bellman equation formulation in terms of the function Q, iteration principle of greedy policy and function approximation. It also presents the design method of the online optimal control system, that is based on the adaptive critic approach, where besides the gains of the DLQR controller are self-adjustable, the computing of the function Q is fully independent of plant model. The simulation results aiming at the optimal control and stabilization of the 3-DOF helicopter and that verify the performance of the RLS methods for the approximation of the action value function of the DLQR for the proposed algorithm in this work are presented in Section 4. Finally, the conclusions and commentaries are contained in Section 5.

2. SYSTEM DESCRIPTION

The experimental platform used in this research is a three degree-of-freedom (3-DOF) helicopter, whose assembly is depicted in Figure 1, and in Figure 2 one can observe the schemes of the helicopter.

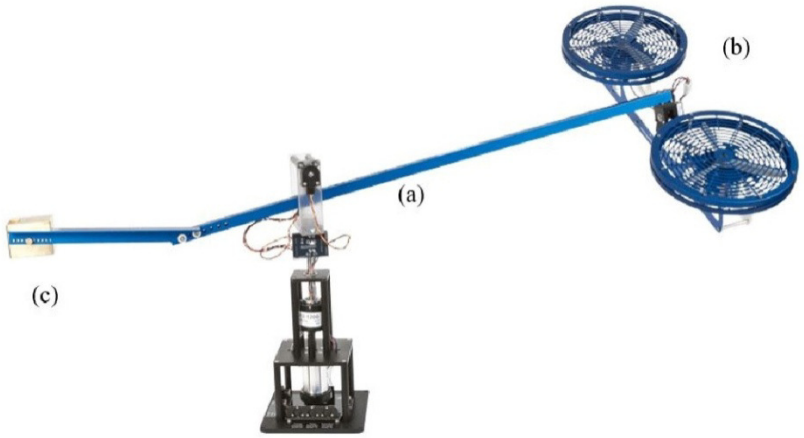

Figure 1. Configuration of the helicopter experimental system and its components: (a) main beam, (b) double rotor and (c) counterweight.

The 3-DOF helicopter is assembled over a stable basis and its primary components are the main beam, a set of a double rotor and the counterweight. The main beam is assembled in a way that allows the rotor assembly to rotate in continuous circles. This rotation movement is called travel. It occurs over a vertical axis which goes through a slip-ring and is perpendicular to the basis. In the bearing and slip-ring assembly there is a pivot point that allows the main beam to raise and lower. This movement is described as pitch or elevation, and it occurs about an axis which is parallel to the basis. In the longer end of the main beam, there is another bearing whose axis is parallel to the beam. It allows a set of double rotors driven by DC motors to rotate around that bearing. The rotational movement of the rotors is referred to as roll, and it occurs around an axis that goes through the main beam. The motors may provide collective or differential (cyclic) voltage. The collective voltage generates the elevation movement of the main beam, and the differential voltage generates the roll movement. The roll movement of the rotors, in turn, originates the travel movement of the assembly. In the other end of the main beam, there is a

counterweight that reduces the energy requirements in the motors, reducing the effective weight of the rotor assembly.

The dynamic of the helicopter can be described by a sixth-order nonlinear model, and for the derivation of the equations of the system, a system of coordinates with its origin in the set of bearing and slip-ring is used, being the travel (ψ) the circular movement of the main beam, roll (ϕ) the movement of the set of rotors, and elevation (θ) the up and down movement of the beam, as well as its respective velocities (travel velocity ($\dot{\psi}$), roll velocity ($\dot{\phi}$), and elevation velocity ($\dot{\theta}$)). The corresponding angles are shown in Figure 2.

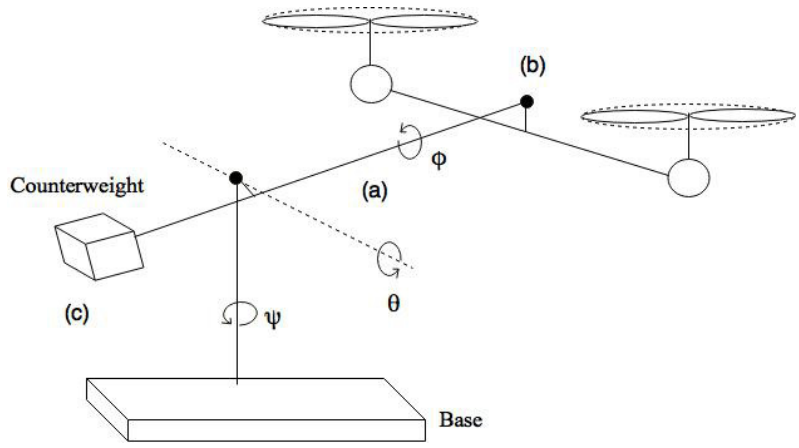

Figure 2. Configuration of the helicopter experimental system.

In this way, the state equation can be expressed as

$$\dot{x} = f(x, u) \tag{1}$$

in which $x = \left[\phi\ \dot{\phi}\ \theta\ \dot{\theta}\ \psi\ \dot{\psi}\right]^T$ and $u = [u_1 u_2]^T$, where the control variables $u_1$and $u_2$corresponding to the voltages given to the left and right motors, respectively.

To simplify the hypothesis, it is considered that the inertia of the helicopter can be represented bypoint masses associated with the body of the helicopter, the counterweight, the gravity center of the sustentation bundle, and with the mass position of active disturbance. Besidesthat, there are the viscous friction effects in the equations that describe the dynamic of the pitchand travel movements, aiming to make it more realistic for the simulation,disregarding the airresistance and the angular moment of the propellers that rotate in the same direction. The forcesgenerated by the propellers do not depend on the relative movement of the body of the helicopter regarding the air. Ultimately, the electromechanical dynamic of the motor-propollers sets is disregarded,which is much faster than the movement dynamic of the system as a whole.

The 3-DOF helicopter has been an object of study of several works in the literature, such as [18 – 21]. Among the proposed models to represent the dynamic of the helicopter, the one presented in[19] was used here. Such model was obtained through the formalism of Lagrange, being nonlinearand sixth-order. Through some mathematical simplifications, this model can be expressed by thefollowing set of differential equations:

$$\dot{x}_1 = x_2$$
$$\dot{x}_2 = \xi_{16} \cdot \{\xi_1(u_1^2 - u_2^2) + \xi_2(u_1 - u_2) - v_2 \cdot x_2\}$$

$$\begin{aligned}
\dot{x}_3 &= x_4 \\
\dot{x}_4 &= x_6^2 \cdot \{\xi_3 \cdot \sin(2x_3) + \xi_4 \cdot \cos(2x_3)\} + \xi_5 \cdot \sin(x_3) + \xi_6 \cdot \cos(x_3) \\
&\quad + \{\xi_7(u_1^2 + u_2^2) + \xi_8(u_1 + u_2)\} \cos(x_1) \\
\dot{x}_5 &= x_6 \\
\dot{x}_6 &= \{\xi_{13} + \xi_{14} \cdot \sin(2x_3) + \xi_{15} \cdot \cos(2x_3)\}^{-1} \\
&\quad \cdot \{v_1 - v_3 \cdot x_6 + [\xi_9(u_1^2 + u_2^2) + \xi_{10}(u_1 + u_2)] \cdot \sin(x_1) + x_4 \cdot x_6 \cdot [\xi_{11} \\
&\quad \cdot \sin(2x_3) + \xi_{12} \cdot \cos(2x_3)]\}
\end{aligned} \tag{2}$$

in which x_1,$x_3$and $x_5$represent the roll, elevation and travel angles (in rad), x_2,$x_4$and $x_6$represent their respective derivatives (rad/s), and $u_1$and $u_2$represent the input voltages of the left andright motors. The other parameters are constants related to the physical dimensions and masses ofthe various components of the helicopter, as well as the constants related to the viscous friction,and their used values are presented in Appendix A.

3. ADHDP FOR ONLINE OPTIMAL CONTROL

The Action-Dependent Heuristic Dynamic Programming (ADHDP) framework for online designof discrete linear quadratic regulator (DLQR) control systems is presented in this section. It isdescribed how to implement an optimal adaptive control using reinforcement learning guided bygreedy iteration schemes and function approximation methods to obtain optimal decision policies online in real-time. Qlearning is a reinforcement learning method that results in an adaptivecontrol algorithm for optimal control solution for completely unknown systems. The parametrizations of Bellman's equation, the utility function and the dynamic system assemble the frameworkof online optimal control design, where the DLQR control policy is estimated online in real-timedirectly from data observed along the system trajectories, this means that the controller proposedsolve the Riccati equation without knowing the system matrix.

3.1. Bellman-DLQR Problem Formulation

The models of the dynamic system f and of the control policy h are linear mappings that arerepresented for combiners of the states and inputs. The state $f(x_k, u_k)$ and decision policy $h(x_k)$parameterizations [22] are given by

$$f(x_k, u_k) = Ax_k + Bu_k \tag{3}$$

and

$$u_k = h(u_k) = -Ku_k \tag{4}$$

where $A \in \Re^{n \times n}$, n is the order of the system, $x \in \Re^n$ is the state, $B \in \Re^{n \times n_e}$, n_e is the amount of control inputs, $u \in \Re^{n \times n_e}$ is the control input, and $K(\cdot) \in \Re^{n_e \times n}$ is the matrix of state feedback gains.

The utility function r associated with the system (3)-(4) has a quadratic form that is given by

$$r(x_k, u_k) = x_k^T Q x_k + u_k^T R u_k \tag{5}$$

with weighting matrices $Q = Q^T \geq 0$ and $R = R^T > 0$ symmetric [23].

For the DLQR control design, the parametrizations of the utility function, Eq.(5), and decision(control) policy, Eq.(4), are replaced in the state-value function $V^h(x_k) = \sum_{i=k}^{\infty} \gamma^{i-k} r(x_i, h(u_i))$to obtain the parameterized DLQR cost function.

$$V^h(x) = r\big(x, h(x)\big) + \gamma V^h f\big(x, h(x)\big) \tag{6}$$

where $0 < \gamma \leq 1$ is the discount factor. So, the goal is to establish an control or decision policyh^*that minimizes the discounted sum of the instantaneous costs, which satisfies the inequality$V^{h^*}(x) \leq V^h(x)$. According to Bellmman's optimality principle, the optimal cost V^*satisfiesthe discrete time HJB equation [24], as follows

$$V^*(x) = \min_{h(\cdot)}\{r\big(x, h(x)\big) + \gamma V^* f\big(x, h(x)\big)\} \tag{7}$$

3.2. DLQR based on$Q$ Learning

The action-dependent heuristic dynamic programming (ADHDP) approach is based on the Qlearning, which consists of a model free method that estimates the function Q for any optimalor non-optimal policy [25, 26] based only the state transition cost samples of the instantaneouscost function. The function Q, or the action-value function, is defined as [27]

$$Q^h(x_k, u_k) = r(x_k, u_k) + \gamma V^h f(x_k, u_k) \tag{8}$$

From the equation (8), it can be seen that given a fixed policy$V^h(x) = Q^h\big(x, h(x)\big)$.Thus, thefunction Q and the optimal function Q^*can be expressed in the Bellman form by

$$Q^h(x_k, u_k) = r(x_k, u_k) + \gamma Q^h(x_{k+1}, h(x_{k+1})) \tag{9}$$

$$Q^*(x_k, u_k) = r(x_k, u_k) + \gamma Q^*(x_{k+1}, h(x_{k+1})) \tag{10}$$

where the optimal control policy is given by

$$h^*(x_k) = \arg\min_{u_k} Q^*(x_k, u_k) \tag{11}$$

In [28] and [23], it is possible to see that for the DLQR, the function Q is quadratic in terms of z, that is,

$$Q(x_k, u_k) = z_k^T S z_k \tag{12}$$

where $z_k^T = [x_k^T \quad u_k^T]$, and S is the learning matrix associated with the function Q, which is given by

$$S = \begin{bmatrix} S_{xx} & S_{xu} \\ S_{ux} & S_{uu} \end{bmatrix} \tag{13}$$

where $S \in \Re^{(n+n_e)\times(n+n_e)}$and the matrices S_{xx}, S_{xu}, S_{ux} e S_{uu} represent the weightings of the state x and the control policy u.

The minimization of the parametrized function Q provides the means to determine the optimalpolicy u^*. The gradient equation$\partial Q/\partial u = 0$, when solved for u, gives the optimal policy u^*,which is described by

$$u^* = K(S)x \tag{14}$$

where $K(S) = -S_{uu}^{-1}S_{ux}$

The parametrization $Q(S)$ induces a policy parametrization $h(S)$. According to the parametrization$u = h(x, \kappa)$ of the control policy, the parameters vector κ is a vectorization of the gain matrix$K$, i.e.$\kappa = vec(K)$ [23].

3.3. RLS-ADHDPApproximation

The method for online optimal control design that will be described in this section is based on theadaptive critic approach and ADHDP algorithms [28, 29]. Such algorithms are developed in thecontext of online DLQR control design that provide the solution of the Riccati equation and theRiccati optimal gain K using greedy iteration schemes.

Considering the quadratic form (12) and (13), assume that, for nonlinear systems, the function Q is parameterized as

$$Q(x, u) = \varphi^T(z)\theta \tag{15}$$

for some unknown weight vector $\boldsymbol{\theta} = \left[\theta_1\theta_2 \ \dots \ \theta_{n_\theta}\right]^T$, where $n_\theta = (n + n_e)(n + n_e + 1)/2$ corresponds to the number of parameters to be estimated, and a basis functions vector $\boldsymbol{\varphi}(z) = \left[\phi_1(z)\,\phi_2(z) \ \dots \ \phi_{n_\theta}(z)\right]^T$, with $z_k = [x_k^T \ \ u_k^T]^T$.For the parameter vector $\boldsymbol{\theta}$ estimation problem, the recursive least squares (RLS) method is considered. Such approach aims at carrying out online learning for optimal control via cost function estimatives of a given policy, constructed from the data $(z_k, z_{k+1}, r(x_k, u_k))$, which are observed throughout the system trajectory.

The criteria used in the policy evaluation based on optimization make the Q learning structure a critic adaptive scheme where the policy iteration step (critic network) determines the least squares solution to $\boldsymbol{\theta}_{k+1}$

$$(\boldsymbol{\varphi}^T(z_k) - \gamma\boldsymbol{\varphi}^T(z_{k+1}))\boldsymbol{\theta_k} = x_k^T Q x_k + u_k^T R u_k \tag{16}$$

and the policy improvement step (action network) determines an improved policy that is given by

$$h_{k+1}(x_k) = \arg\min_{h(\cdot)}(\phi^T(z_k)\boldsymbol{\theta}_k) \tag{17}$$

where each pair of weightings Q and R defines a different controller. Therefore, exploring the possible weighing space, one approximates the dynamic programming solution for the optimal controller. ADHDP is a method which improves the controller from an iteration to the next, from the time instant k until the instant $k + 1$ [29].

The matrix vectorization and the Kronecker product theory [13, 30] contribute for an approximate solution of the HJB-Riccati equation obtained through an iterative scheme such as

$$\phi^T(z_k)\boldsymbol{\theta}_k = r(x_k, u_k) \tag{18}$$

where $\boldsymbol{\theta}$ is the parameter vector corresponding to the matrix S vectorization, $\phi^T(z_k)$ is the regression vector and $r(x_k, u_k)$ is the target, which are given by

$$\begin{aligned}\phi_k &= z_k^T \otimes z_k^T \\ &= [z_{1,k}^2; z_{1,k}z_{2,k}; z_{2,k}^2; \dots; z_{n-1,k}z_{n,k}; z_{n,k}^2]^T -\end{aligned} \tag{19}$$

$$\gamma[z_{1,k+1}^2; z_{1,k+1}z_{2,k+1}; z_{2,k+1}^2; \dots; z_{n-1,k+1}z_{n,k+1}; z_{n,k+1}^2]^T$$

$$r(x_k, u_k) = x_k^T Q x_k + u_k^T R u_k \tag{20}$$

The problem consists in determining a parameter vector $\boldsymbol{\theta}$ estimative from a set of pairs of observations and regressors $\{(d(\cdot), \phi_k, k = 1, 2, \dots, N\}$, taking in accountonline designs for real-time applications. The least squares estimative of $\boldsymbol{\theta}$ is defined as the vector that minimizes the following cost function

$$J(\theta, N) = \sum_{k=1}^{N} \mu^{N-k} [r(x_k, u_k) - \phi_k^T \boldsymbol{\theta}]^2 \tag{21}$$

where μ is the forgetting factor, $0 < \mu \leq 1$, and N is the number ofsample data.
The least-squares solution of the problem (21) is given by [31]

$$\boldsymbol{\theta}_N = \boldsymbol{\Phi}_N^{-1} \boldsymbol{\eta}_N \tag{22}$$

where$\boldsymbol{\Phi}_N = \sum_{k=1}^{N} \mu^{N-k} \phi_k \phi_k^T$is the correlation matrix, $n_\theta \times n_\theta$, of the input data vector ϕ, and $\boldsymbol{\eta}_N = \sum_{k=1}^{N} \mu^{N-k} \phi_k r(x_k, u_k)$ is the crossed correlation vector between the LS estimator inputs and the desired response.

Through algebraic manipulations, Eq.(22) is developed in a recursive form given by

$$\boldsymbol{\theta}_k = \boldsymbol{\theta}_{k-1} + \boldsymbol{\Phi}_k^{-1} \phi_k \big(r(x_k, u_k) - \phi_k^T \boldsymbol{\theta}_{k-1}\big) \tag{23}$$

where $\boldsymbol{\Phi}_k$ is a recursive form given by

$$\boldsymbol{\Phi}_k = \mu \boldsymbol{\Phi}_{k-1} + \phi_k \phi_k^T \tag{24}$$

Applying the matrix inversion lemma to the Eq.(24), the RLS estimation in the forms (23)-(24) can be rewritten as

$$\boldsymbol{\theta}_k = \boldsymbol{\theta}_{k-1} + \boldsymbol{L}_k \big(r(x_k, u_k) - \phi_k^T \boldsymbol{\theta}_{k-1}\big) \tag{25}$$

where

$$\boldsymbol{L}_k = \boldsymbol{\Gamma}_k \phi_{\boldsymbol{k}} = \frac{\boldsymbol{\Gamma}_{k-1} \phi_{\boldsymbol{k}}}{\mu + \phi_k^T \boldsymbol{\Gamma}_{k-1} \phi_{\boldsymbol{k}}} \tag{26}$$

and

$$\boldsymbol{\Gamma}_k = \mu^{-1} (\boldsymbol{\Gamma}_{k-1} - \boldsymbol{L}_k \phi_k^T \boldsymbol{\Gamma}_{k-1}) \tag{27}$$

where the matrix Φ, $n_\theta \times n_\theta$, is the inverse of correlation matrix/covariance matrix, and $L_k$is thegain vector, $n_\theta \times 1$.

The ADHDP algorithm main core is developed according to the Eqs.(18)-(22) for theonline implementation based on RLS. The observed data throughout the system trajectoryis$(z_k, z_{k+1}, r(x_k, u_k))$ with $z_k = [x_k^T \quad u_k^T]^T$. The vector$z_{k+1} = [x_{k+1}^T \quad u_{k+1}^T]^T$is computed using$u_{k+1} = h_k(x_{k+1})$ where$h_k(\cdot)$ is the current policy. A probing noise mustbe added to the control input as a conditionto obtain the persistence of excitation,

whichis$u_k = -Kx_k + \epsilon_k$[22]. That scheme is capable ofsolving the HJB-Riccati equation online with noknowledge of the system dynamics.

3.3.1. RLS$_\mu$ – ADHDP-DLQR Algorithm

TheRLS$_\mu$-ADHDP-DLQR algorithm has two main blocks that are responsible for the policyevaluation and policy improvement steps to determine the control policy based on reinforcement learning methods. In the block of policy evaluation, Kronecker product (steps 25-27) and theequations of the RLS estimator (step 28) are executed so the matrix S is approximated, followedby the policy improvement (steps 31-32) block, the latter is a greedy policy with respect to theapproximation $S^{\theta_{k+1}}$. The block 0 contains the fixed parameters of the system inherent to theoptimization problem, which are the weighing matrices Q and R, the matrices A and B of thedynamic system, and the discount fator γ. The forgetting factor μ, the parameter θ, and thematrix$\boldsymbol{\Gamma}$ are the necessary conditions for the RLS estimation problem.

ALGORITHM 1 - RLS$_\mu$-ADHDP-DLQR

---

1 ►**Block 0 – Initialization**
2 ►Weighting and Dynamic System Matrices – Q, R, A_d, B_d
3 ►Initial Policy – K_0
4 ►State Resetting – x_{revit}, n_{revit}
5► Initial State – x_0
6 ►Discount Factor – $0 < \gamma \leq 1$
7 ►RLS$_\mu$Parameters:θ_0,Γ_0
8 ►Forgetting Factor **–** $0 < \mu \leq 1$
9 ►Iteration Number – N.

---

10 ►**-** Iterative Process
11 ► **for**$k \leftarrow 0 : N$
12 **do**
13 ►**Block 1– Environment Simulation**
14 Control Noise (Probing noise of control signal)
15 $\epsilon_k \leftarrow [\ ]$
16 Control Action
17 $u_k \leftarrow -K_k x_k + \epsilon_k$
18 States
19 $x_{k+1} \leftarrow A_d x_k + B_d u_k$
20 Next Control Action
21 $u_{k+1} \leftarrow -K_k x_{k+1}$

---

22 ►**Block 2– Approximate Policy Evaluation**
23►Target Assembling
24 $r(x_k, u_k) \leftarrow x_k^T Q x_k + u_k^T R u_k$
25 ►Basis Set - Kronecker Product
26 $\phi_k = [z_{1,k}^2; z_{1,k}z_{2,k}; z_{2,k}^2; \dots;\ z_{n-1,k}z_{n,k}; z_{n,k}^2]^T - \gamma[z_{1,k+1}^2;$
27 $\dots; z_{1,k+1}z_{2,k+1}; z_{2,k+1}^2; \dots;\ z_{n-1,k+1}z_{n,k+1}; z_{n,k+1}^2]^T$
28 ►Recursive least-square : Update$\boldsymbol{\theta}_{k+1}$ via RLS recurrence (25)-(27)
29 ►Smatrix recovery from vectorθ
30 $$S^{\theta_{k+1}} \leftarrow \begin{bmatrix} \theta_1 & \theta_2/2 & \cdots & \theta_{(n+n_e)}/2 \\ \theta_2/2 & \theta_{(n+n_e)+1} & \cdots & \theta_{2(n+n_e)-1/2} \\ \vdots & \vdots & \ddots & \vdots \\ \theta_{(n+n_e)}/2 & \theta_{2(n+n_e)-1}/2 & \cdots & \theta_{n_\theta} \end{bmatrix}$$

---

31 ► **Block 3– Policy Improvement** (Feedback Optimal Gain K)
32 $K_{k+1} \leftarrow (S_{uu}^{\theta_{k+1}})^{-1}(S_{ux}^{\theta_{k+1}})$

---

33 **if**$i\%n_{revit} = 0$
34 **then**

35 $x_{k+1} \leftarrow x_{revit}$

36 **End** – Iterative Process

4. SIMULATIONRESULTS

In this section, the simulation results of the online optimal control design for the 3-DOF Helicopter system are presented, using an adaptive critic scheme based on greedy policy iterationtechnique.The typical parameters of configuration of the 3-DOF helicopter system are presented in AppendixA. The controller design via ADHDP is established aiming at maintaining the stabilization of the3-DOF helicopter system in real-time, taking into account the iterative process configuration , aswell as the reference policy, which is established offline by Schur's solution [32] for the HJB-Riccati equation.

The linearization for the adopted operation point of the nonlinear model of the helicopter wascarried out using the parameters given in Appendix A. The matrices of the linear continuousmodel are expressed as:

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & -0.753 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ -0.098 & 0 & -1.192 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ -1.257 & 0 & 0 & 0 & 0 & -0.457 \end{bmatrix} \tag{28}$$

And

$$B = \begin{bmatrix} 0 & 0 \\ 2.814 & -2.814 \\ 0 & 0 \\ 0.394 & 0.394 \\ 0 & 0 \\ -0.035 & -0.035 \end{bmatrix} \tag{29}$$

The chosen output variables were the roll, elevation and travel angles ($y = [\phi\theta\ \psi]^{\mathrm{T}}$). Besidesthat, since there is no direct transmission between inputs and outputs of the plant, the C and Dmatrices were defined as

$$C = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix} \text{and } D = [0] \tag{30}$$

For many executed simulations, the influence of the forgetting factor μ can be verified in theconvergence process for the solution of the HJB-Riccati equation with different values of μ. Inthis approach, the algorithm presents a revitalization condition due to null-state problems that leadthe matrix of regressors to a null rank [7]. So, it is necessary to perform the system revitalizationafter each interval n_{revit}, which is given by the dimension of the regression vector ϕ_k.

For the implementation of the RLS$_\mu$-ADHDP-DLQR algorithm the initial conditions and systemparameters are: an admissible initial policy K_0; discount fator γ; initial state of the system$x_0 = [0.18\ -0.19\ -0.32\ 0.61\ -0.5]^T$ for the angles in radians and quadratic matrices of the cost function respectively given by $Q = [0.1\ 0\ 0\ 0\ 0\ 0;\ 0\ 1\ 0\ 0\ 0\ 0;\ 0\ 0\ 1\ 0\ 0\ 0;\ 0\ 0\ 0\ 0.01\ 0\ 0;\ \ldots;\ 0\ 0\ 0\ 0\ 1\ 0;\ 0\ 0\ 0\ 0\ 0\ 1]; R =$

$[10^2\ 0;0\ 10^2]$. The parameters that initialize the proposed RLSestimator are given by vector $\boldsymbol{\theta}_0 = 10^3 * [1\,0\,0\,0\,0\,0\,0\,0\,1\,0\,0\,0\,0\,0\,0\,0\,1\,0\,0\,0\,0\,0\,0\,0\,1\,0\,0\,0\,0\,0\,0\,0\,1\,0\,0\,0\,\ldots 1\,0\,0\,1\,0\,1]^T$; and matrix $\Gamma_0 = 10 * I_{36\times36}$, where I is na identity matrix of order n_θ. The matrices of the discretized system for the sampling interval $T = 0.1$ are obtainedthrough the zero-orderhold (ZOH) method. For the simulations the following values of μ are considered: 0.7640 and0.9802.

4.1. Forgetting Factor $\mu = 0.764$

The evolution of the iterative process of the Q-function estimation by the $\mathrm{RLS}_\mu - \mathrm{ADHDP} - \mathrm{DLQR}$algorithm is presented in Figure 3 for a cycle of 5000 iterations, considering the forgetting factor $\mu = 0.764$. The curves (a)-(f) of Figure 3 represent the convergence behavior of the elements s_{11}, s_{26}, s_{33}, s_{55}, s_{74} and s_{87} of the matrix S corresponding to the componentes θ_1, θ_{13}, θ_{16}, θ_{27}, θ_{25}andθ_{35} of the parameter vector θ, respectively. There was a quick convergence, without large oscillations during the transitory period, which was achieved after only 1600 iterations.

The control strategy was adopted to solve the regulation problem of the helicopter, that is to maintain the 3-DOF helicopter in a steady and pre-stablished flight condition, considering restrictionsof the helicopter angles (which can be interpreted as obstacles that limit its maneuvering space),restrictions of control variables and external disturbance.

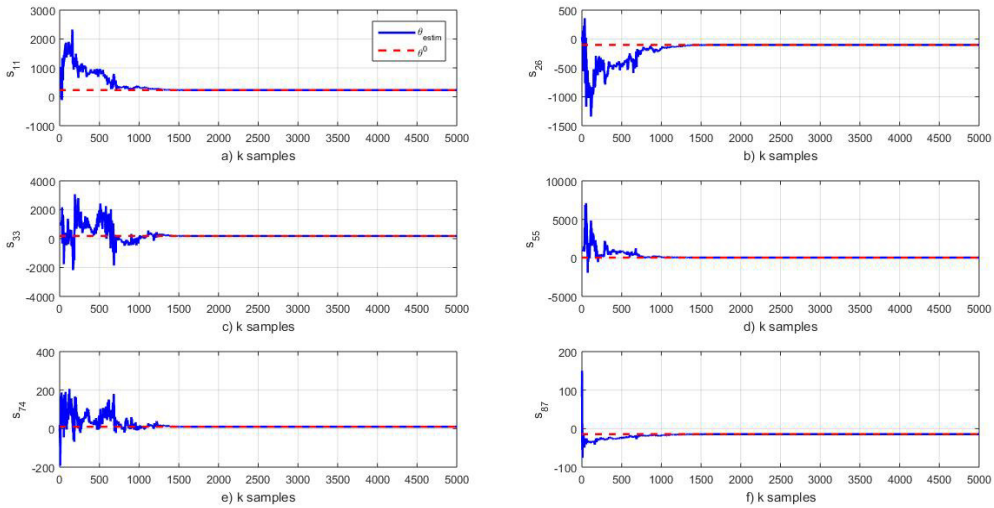

Figure 3.Evolution of the iterative process for the parameterss_{11}, s_{26}, s_{33}, s_{55}, s_{74} and s_{87} for 5000 cycle of iterations, with the forgetting fator $\mu = 0.764 - \mathrm{RLS}_\mu - \mathrm{ADHDP} - \mathrm{DLQR}$

Figures 4 and 5 correspond to the control effort and the state trajectories of the system,respectively,from the action-environment-observation interactivity of the dynamic system simulator for the$\mathrm{RLS}_\mu - \mathrm{ADHDP} - \mathrm{DLQR}$ algorithm.

From Figure 5, one can observe that the values of the states concentrate in a range that correspondsto the equilibrium point of the system. The recurring variations of the control effort in Figure 4indicate the effort at regulating, the states in a quick way, so that the plant of the 3-DOF Helicopter is in conformity with the pre-established design.

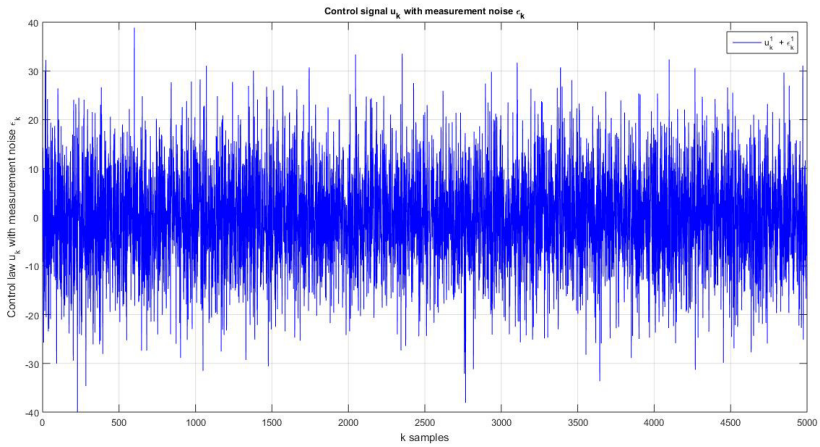

Figure 4.Control signal with tracking noise

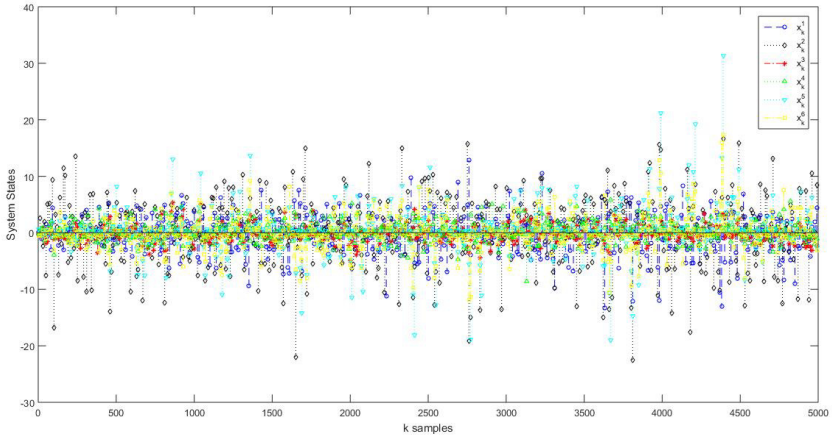

Figure 5.States x_1, x_2, x_3, x_4, $x_5$andx_6 of the system for some samples

4.2. Forgetting Factor $\mu = 0.9802$

For a cycle of 5000 iterations, the evolution of the iterative process of the Q-function estimation by the $\text{RLS}_\mu - \text{ADHDP} - \text{DLQR}$algorithm is presented in Figure 6, considering the forgetting factor $\mu = 0.980$. The curves (a)-(f) of Figure 6 represent the convergence behavior of the elements s_{11}, s_{26}, s_{33}, s_{55}, s_{74} and s_{87} of the matrix S correspondents to the componentes θ_1, θ_{13}, θ_{16}, θ_{27}, θ_{25}andθ_{35} of the parameter vector θ, respectively. There was a subtle convergence, without large oscillations during the transitory period, which was achieved after only 1600 iterations.The same smooth convergence that took place for thefactor $\mu = 0.764$also happened to the factor $\mu = 0.9802$, thedifference between the two factors is the accommodation time. One can observe throughout thesimulations that for higher factors the convergence time tends to grow. For the forgetting fator $\mu = 0.9802$, the convergence was achieved after only 2500 iterations.

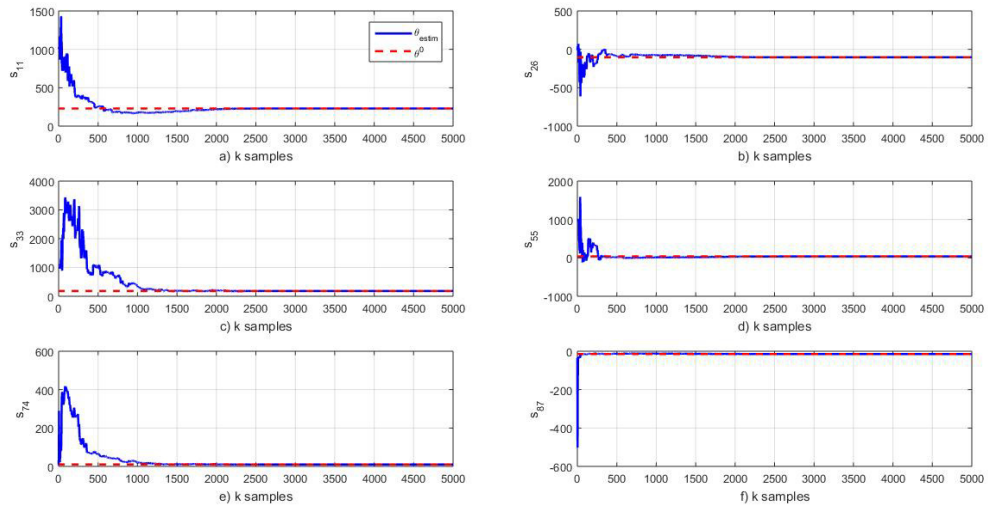

Figure 6.Evolution of the iterative process for the parameterss_{11}, s_{26}, s_{33}, s_{55}, s_{74} and s_{87} for 5000 cycle of iterations, with the forgetting fator $\mu = 0.764 - \mathrm{RLS}_{\mu} - \mathrm{ADHDP} - \mathrm{DLQR}$

As for the previous factor, we have in Figures 7 and 8, respectively, the control effort added to thetracking noise and the trajectories of the system states. The chosen operating point remains thesame, corresponding to the situation in which the helicopter finds itself still with elevation angle 14degrees below horizontal position. Suppose that the changes in the forgetting factor (disturbances)affect the input variables of the system, as one can see in Figure 8.

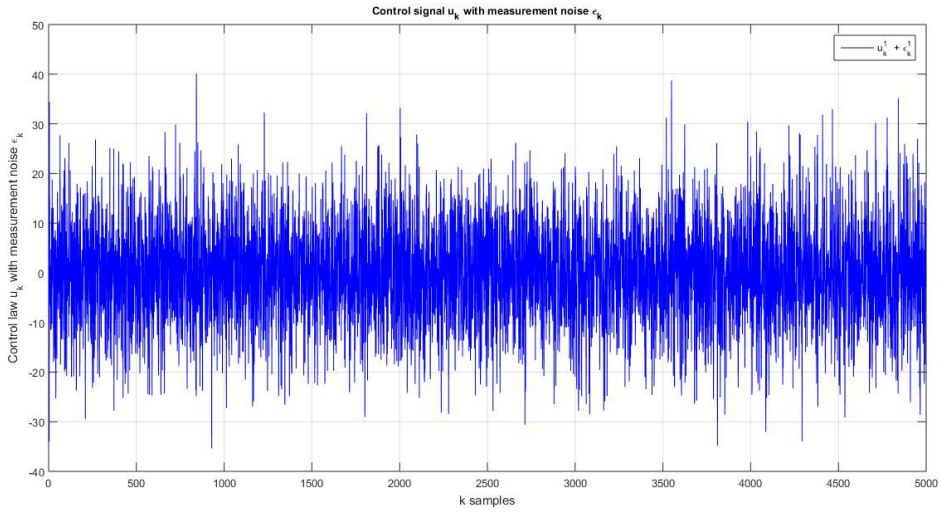

Figure 7.Control signal with tracking noise

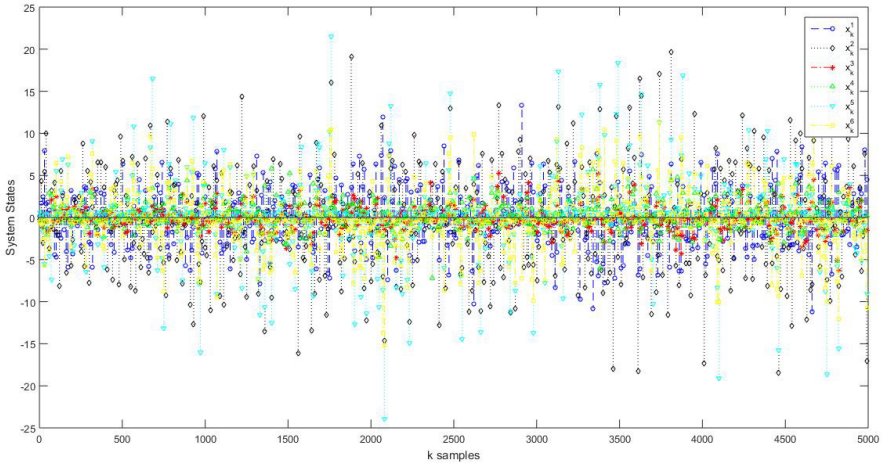

Figure 8.States x_1, x_2, x_3, x_4, $x_5$andx_6 of the system for some samples

The ADHDP methodology via process of RLS estimation of the optimal DLQR decision policies represent a method that performs the RLS approximation of the solution of the HJB-Riccati equation in the context of adaptive optimal control project in real-time based on the structure of learning by effort over a greedy policy iteration. Such methodology seems pretty successfulregarding the control and stabilization of the system of the 3-DOF helicopter, being that form a reference for the study and control of systems whose mathematical model is complex for the controlprocess in real-time.

5. SIMULATIONRESULTS

In the present design, the adaptive critic method, based on greedy policy iterations, wasemployedfor obtaining solutions of online optimal control problems of discrete-time nonlinear systems,especially the control and stabilization problem of a 3-DOF helicopter.

First of all, a mathematical model was adopted for the 3-DOF helicopter system for getting aprocess simulator for the ADHDP-DLQR design purpose. In the performance evaluation of theRLS$_\mu$ − ADHDP − DLQRalgorithm, one can verify with respect to the selection of the forgettingfactor, its large influence in the convergence process for the solution of the HJB-Riccati equation.The aim of the proposed methodology is to improve the ADHDP approach performance via RLS estimation of the DLQR optimal decision policies.

The results were promising for the multivariable dynamic system models, since the appropriatechoice of the forgetting factor considerably improves the performance of the RLS$_\mu$ − ADHDP − DLQR method. The use of UDU^Tfactorization, as well as of the $Q\mathcal{R}$ decomposition used inthe critic net of the algorithm are one of our main research topics in the future, aiming to overcome problems inherent to the RLS estimation such as the numeric stability loss of the covariancematrix, causing a possible convergence loss of the systems under study.

ACKNOWLEDGEMENTS

The authors are indebted to the State University of Maranhão (UEMA), Postgraduate Programin Computer Engineering and Systems (PECS), Federal University of Maranhão (UFMA),Technological Institute of Aeronautics (ITA), and Foundation for Support on Research and ScientificDevelopment of Maranhão (FAPEMA).

A. APPENDIX

Table 1. Parameter values

Parameter	Value	Parameter	Value
ξ_1	0,1117 N/V^2	ξ_{11}	1,0567 kg·m^2
ξ_2	0,0449 N/V	ξ_{12}	-0,2515 kg·m^2
ξ_3	-0,4843	ξ_{13}	0,5454 kg·m^2
ξ_4	0,1153	ξ_{14}	0,1258 kg·m^2
ξ_5	-1,0389 N/(m·kg)	ξ_{15}	0,5283 kg·m^2
ξ_6	-1,3170 N/(m·kg)	ξ_{16}	4,1832 (m·kg)$^{-1}$
ξ_7	0,0656 N/(m·kg·V^2)	v_1	0,107 N·m
ξ_8	0,0264 N/(m·kg·V)	v_2	0,18 N·s
ξ_9	-0,0718 N·m/V^2	v_3	0,47 N·m·s
ξ_{10}	-0,0289 N·m/V	-	-

REFERENCES

[1] F. L. Lewis and D. Liu, Reinforcement learning and approximate dynamic programming for feedback control. John Wiley & Sons, 2012, vol. 17

[2] G. G. Lendaris, "A retrospective on adaptive dynamic programming for control," in NeuralNetworks, 2009. IJCNN 2009. International Joint Conference on. IEEE, 2009, pp. 1750–1757.

[3] P. J. Werbos, "Advanced forecasting methods for global crisis warning and models of intelligence", General Systems Yearbook, vol. 22, no. 12, pp. 25–38, 1977.

[4] "A menu of designs for reinforcement learning over time," Neural networks forcontrol,pp. 67–95, 1990.

[5] A. Al-Tamimi, F. L. Lewis, and M. Abu-Khalaf, "Discrete-time nonlinear hjb solution usingapproximate dynamic programming: Convergence proof", IEEE Transactions on Systems,Man, and Cybernetics, Part B (Cybernetics), vol. 38, no. 4, pp. 943–949, 2008.

[6] D. Liu and Q. Wei, "Policy iteration adaptive dynamic programming algorithm for discretetime nonlinear systems", IEEE Transactions on Neural Networks and Learning Systems,vol. 25, no. 3, pp. 621–634, 2014.

[7] E. F. Ferreira, P. H. Rêgo, and J. V. Neto, "Numerical stability improvements of state-valuefunction approximations based on rls learning for online hdp-dlqr control system design", Engineering Applications of Artificial Intelligence, vol. 63, pp. 1–19, 2017.

[8] X. Xu, H.-g. He, and D. Hu, "Efficient reinforcement learning using recursive least-squaresmethods", Journal of Artificial Intelligence Research, vol. 16, pp. 259–292, 2002.

[9] B. Chu, D. Hong, J. Park, and J.-H. Chung, "Passive dynamic walker controller design employing an rls-based natural actor–critic learning algorithm", Engineering Applications ofArtificial Intelligence, vol. 21, no. 7, pp. 1027–1034, 2008.

[10] Y. Cheng, H. Feng, and X. Wang, "Efficient data use in incremental actor–critic algorithms", Neurocomputing, vol. 116, pp. 346–354, 2013.

[11] O. Pietquin, M. Geist, and S. Chandramohan, “Sample efficient on-line learningof optimal dialogue policies with kalman temporal differences”, in Proceedingsof the Twenty-Second international joint conference on Artificial Intelligence, ser.IJCAI’11, vol. 3. AAAI Press, 2011, pp. 1878–1883. [Online]. Available: http://dx.doi.org/10.5591/978-1-57735-516-8/IJCAI11-314

[12] M. Geist, O. Pietquin, and G. Fricout, “Kalman temporal differences: The deterministic case”, in Adaptive Dynamic Programming and Reinforcement Learning, 2009. ADPRL ’09.IEEE Symposium on, 30 2009-april 2 2009, pp. 185–192.

[13] P. H. M. Rêgo, J. V. d. F. Neto, and E. M. Ferreira, “Convergence of the standardrls method and udu t factorisation of covariance matrix for solving the algebraic riccatiequation of the dlqr via heuristic approximate dynamic programming”,InternationalJournal of Systems Science, vol. 46, no. 11, pp. 2006–2028, 2013. [Online]. Available:http://dx.doi.org/10.1080/00207721.2013.844283

[14] C. J. C. H. Watkins, “Learning from delayed rewards”, Ph.D. dissertation, King’s College,Cambridge, 1989.

[15] C. J. Watkins and P. Dayan, “Q-learning”, Machine learning, vol. 8, no. 3-4, pp. 279–292,1992.

[16] P. J. Werbos, “Approximate dynamic programming for real-time control and neural modeling”,Handbook of intelligent control, pp. 493–526, 1992.

[17] L. Busoniu, R. Babuska, B. De Schutter, and D. Ernst, Reinforcement learning anddynamicprogramming using function approximators. CRC press, 2010, vol. 39.

[18] M. A. Ishutkina, “Design and implimentation of a supervisory safety controller for a 3dofhelicopter”, Ph.D. dissertation, Massachusetts Institute of Technology, 2004.

[19] R. V. Lopes, “Modelagem e controle preditivo de um helicóptero com três graus deliberdade”, 136f. Dissertations (Mestrado em Sistemas e Controle)-Instituto Tecnológico deAeronáutica, São José dos Campos, 2007.

[20] B. Sapi ´ nski and M. Rosół, “Autonomous control system for a 3 dof pitch-planesuspension model with mr shock absorbers”,Computers& structures, vol. 86, no. 3, pp. 379–385,2008. [Online]. Available: https://doi.org/10.1016/j.compstruc.2007.02.017

[21] R. Breganon, “Controle de arfagem e guinada de um sistema de hélices paralelas”, Ph.D.dissertation, Universidade de São Paulo, 2009.

[22] S. J. Bradtke, B. E. Ydstie, and A. G. Barto, “Adaptive linear quadratic control using policyiteration”, in American Control Conference, 1994, vol. 3. IEEE, 1994, pp. 3475–3479.

[23] J. Fonseca and P. Rêgo, “Qr-tuning and approximate-ls solutions of the hjb equation foronline dlqr design via state and actiondependent heuristic dynamic programming”,International Journal of Innovative Computing, Information and Control (IJICIC)., vol. 10, no. 3,pp. 1071–1094, 2014.

[24] R. Bellman, “Dynamic programming and stochastic control processes”,Information andcontrol, vol. 1, no. 3, pp. 228–239, 1958.

[25] T. Landelius, “Reinforcement learning and distributed local model synthesis”, Ph.D.dissertation, Linköping University Electronic Press, 1997.

[26] F. L. Lewis and D. Vrabie, “Adaptive dynamic programming for feedback control”, in AsianControl Conference, 2009. ASCC 2009. 7th. IEEE, 2009, pp. 1402–1409.

[27] P. E. Stingu and F. L. Lewis, “Adaptive dynamic programming applied to a 6dof quadrotor”, in Computational Modeling and Simulation of Intellect: Current State and Future Perspectives. IGI Global, 2011, pp. 102–130.

[28] F. L. Lewis, D. Vrabie, and K. G. Vamvoudakis, "Reinforcement learning and feedbackcontrol: Using natural decision methods to design optimal adaptive controllers", IEEE Control Systems, vol. 32, no. 6, pp. 76–105, 2012.

[29] Y. Sokolov, R. Kozma, L. D. Werbos, and P. J. Werbos, "Complete stability analysis of aheuristic approximate dynamic programming control design",Automatica, vol. 59, pp. 9–18,2015.

[30] J. Brewer, "Kronecker products and matrix calculus in system theory",IEEE Transactionson circuits and systems, vol. 25, no. 9, pp. 772–781, 1978.

[31] K. J. Åström and B. Wittenmark, Computer-controlled systems: theory and design (3 ed.).Prentice-Hall, 1997.

[32] A. Laub, "A schur method for solving algebraic riccati equations",IEEE Transactions onautomatic control, vol. 24, no. 6, pp. 913–921, 1979.