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

Guilherme B. Sousa1*, Janes V. R. Lima1, Patrícia H. M. Rêgo2, Alain G. Souza3 and João V. Fonseca Neto4

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 optimization 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.
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 markovian paradigms.

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.](image)

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 (ψ), roll velocity (ϕ), and elevation velocity (θ)). The corresponding angles are shown in Figure 2.

In this way, the state equation can be expressed as

\[ \dot{x} = f(x, u) \]  

in which \( x = [ϕ \ ϕ \ θ \ θ \ ψ \ ψ]^T \) 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 by point 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. Besides that, there are the viscous friction effects in the equations that describe the dynamic of the pitch and travel movements, aiming to make it more realistic for the simulation, disregarding the air resistance and the angular moment of the propellers that rotate in the same direction. The forces generated 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-propellers 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 nonlinear and sixth-order. Through some mathematical simplifications, this model can be expressed by the following set of differential equations:

\[
\begin{align*}
\dot{x}_1 &= x_2 \\
\dot{x}_2 &= \xi_1 (u_1^2 - u_2^2) + \xi_2 (u_1 - u_2) - v_2 \cdot x_2
\end{align*}
\]
\[ \begin{align*}
  \dot{x}_3 &= x_4 \\
  \dot{x}_4 &= x_5^2 \cdot (\xi_3 \cdot \sin(2x_3) + \xi_4 \cdot \cos(2x_3)) + \xi_5 \cdot \sin(x_3) + \xi_6 \cdot \cos(x_3) \\
  &\quad + \{\xi_7 (u_1^2 + u_2^2) + \xi_8 (u_1 + u_2)\} \cos(x_1) \\
  \dot{x}_5 &= x_6 \\
  \dot{x}_6 &= \{\xi_{13} + \xi_{14} \cdot \sin(2x_3) + \xi_{15} \cdot \cos(2x_3)\}^{-1} \\
  &\quad \cdot \{v_1 - v_3 \cdot x_6 + [\xi_9 (u_1^2 + u_2^2) + \xi_{10} (u_1 + u_2)] \cdot \sin(x_1) + x_4 \cdot x_6 \cdot [\xi_{11} \\
  &\quad \cdot \sin(2x_3) + \xi_{12} \cdot \cos(2x_3)]\}
\end{align*} \]

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 and right motors. The other parameters are constants related to the physical dimensions and masses of the 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 design of discrete linear quadratic regulator (DLQR) control systems is presented in this section. It is described how to implement an optimal adaptive control using reinforcement learning guided by greedy iteration schemes and function approximation methods to obtain optimal decision policies online in real-time. Q-learning is a reinforcement learning method that results in an adaptive control algorithm for optimal control solutions for completely unknown systems. The parametrizations of Bellman’s equation, the utility function and the dynamic system assemble the framework of online optimal control design, where the DLQR control policy is estimated online in real-time directly from data observed along the system trajectories, this means that the controller proposed solves 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 are represented 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

\[ f(x_k, u_k) = Ax_k + Bu_k \]  
\[ u_k = h(u_k) = -Ku_k \]

where \(A \in \mathbb{R}^{n \times n}\), \(n\) is the order of the system, \(x \in \mathbb{R}^n\) is the state, \(B \in \mathbb{R}^{n \times n}\), \(n_e\) is the amount of control inputs, \(u \in \mathbb{R}^{n \times n}\) is the control input, and \(K(\cdot) \in \mathbb{R}^{n_e \times n}\) 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

\[ r(x_k, u_k) = x_k^T Q x_k + u_k^T R u_k \]

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

\[ V^h(x_k) = \sum_{i=0}^{N} \gamma^{i-k} r(x_i, h(u_i)) \]

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

$$V^*(x) = \min_{h(x)} \{r(x, h(x)) + \gamma V^h(x, h(x))\}$$

(7)

### 3.2. DLQR based on Q Learning

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

$$Q^h(x_k, u_k) = r(x_k, u_k) + \gamma V^h(x_k, u_k)$$

(8)

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

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

(9)

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

(10)

where the optimal control policy is given by

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

(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,

$$Q(x_k, u_k) = z_k^T S z_k$$

(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

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

(13)

where $S \in \mathbb{R}^{(n+n_e) \times (n+n_e)}$ and the matrices $S_{xx}, S_{xu}, S_{ux}, S_{uu}$ 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 optimal policy $u^*$. The gradient equation $\partial Q / \partial u = 0$, when solved for $u$, gives the optimal policy $u^*$, which is described by

$$u^* = K(S)x$$

(14)
The parametrization \[ Q(S) \] induces a policy parametrization \( h(S) \). According to the parametrization \( h = h(x, \kappa) \) of the control policy, the parameters vector \( \kappa \) is a vectorization of the gain matrix \( K \), i.e. \( \kappa \) = \( \text{vec}(K) \) [23].

### 3.3. RLS-ADHDPApproximation

The method for online optimal control design that will be described in this section is based on the adaptive critic approach and ADHDP algorithms [28, 29]. Such algorithms are developed in the context of online DLQR control design that provide the solution of the Riccati equation and the Riccati 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 \( \theta = [\theta_1 \theta_2 ... \theta_{n_\theta}]^T \), where \( n_\theta = (n + n_\kappa)(n + n_\kappa + 1)/2 \) corresponds to the number of parameters to be estimated, and a basis functions vector \( \varphi(z) = [\phi_1(z) \phi_2(z) ... \phi_{n_\theta}(z)]^T \), with \( z_k = [x_k^T \ u_k^T]^T \). For the parameter vector \( \theta \) 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 \( \theta_{k+1} \)

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

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

\[
h_{k+1}(x_k) = \arg \min_{h(\cdot)} (\varphi^T(z_k) \theta_k) \tag{17}
\]

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

\[
\varphi^T(z_k) \theta_k = r(x_k, u_k) \tag{18}
\]

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

\[
\phi_k = z_k^T \otimes z_k^T = [z_{1, k}^2; z_{1, k} z_{2, k}; z_{2, k}^2; ...; z_{n-1, k} z_{n, k}; z_{n, k}^2]^T - \tag{19}
\]
The problem consists in determining a parameter vector $\boldsymbol{\theta}$ estimative from a set of pairs of observations and regressors $\{ (\varphi_k, k = 1, 2, ..., N) \}$, taking into account online designs for real-time applications. The least squares estimative of $\boldsymbol{\theta}$ is defined as the vector that minimizes the following cost function

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

(21)

where $\mu$ is the forgetting factor, $0 < \mu \leq 1$, and $N$ is the number of samples data.

The least-squares solution of the problem (21) is given by [31]

$$\theta_N = \Phi_N^{-1} \eta_N$$

(22)

where $\Phi_N = \sum_{k=1}^{N} \mu^{N-k} \phi_k \phi_k^T$ is the correlation matrix, $n_\theta \times n_\theta$, of the input data vector $\varphi_k$, and $\eta_N = \sum_{k=1}^{N} \mu^{N-k} \phi_k r(x_k, u_k)$ 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

$$\theta_k = \theta_{k-1} + \Phi_k^{-1} \phi_k (r(x_k, u_k) - \phi_k^T \theta_{k-1})$$

(23)

where $\Phi_k$ is a recursive form given by

$$\Phi_k = \mu \Phi_{k-1} + \phi_k \phi_k^T$$

(24)

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

$$\theta_k = \theta_{k-1} + L_k (r(x_k, u_k) - \phi_k^T \theta_{k-1})$$

(25)

where

$$L_k = \Gamma_k \phi_k = \frac{\Gamma_{k-1} \phi_k}{\mu + \phi_k^T \Gamma_{k-1} \phi_k}$$

(26)

and

$$\Gamma_k = \mu^{-1} (\Gamma_{k-1} - L_k \phi_k^T \Gamma_{k-1})$$

(27)

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

The ADHDP algorithm main core is developed according to the Eqs.(18)-(22) for the online implementation based on RLS. The observed data throughout the system trajectory is $\{ (z_k, z_{k+1}, r(x_k, u_k)) \}$ with $z_k = [x_k^T \ u_k^T]^T$. The vector $z_{k+1} = [x_{k+1}^T \ u_{k+1}^T]^T$ is computed using $u_{k+1} = h_k(x_{k+1})$ where $h_k(\cdot)$ is the current policy. A probing noise must be added to the control input as a condition to obtain the persistence of excitation,
which is $u_k = -K_k x_k + e_k$ [22]. That scheme is capable of solving the HJB-Riccati equation online with no knowledge of the system dynamics.

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

The RLS$_{\mu}$-ADHDP-DLQR algorithm has two main blocks that are responsible for the policy evaluation 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 the equations of the RLS estimator (step 28) are executed so the matrix $S$ is approximated, followed by the policy improvement (steps 31-32) block, the latter is a greedy policy with respect to the approximation $S_{\theta_k+1}$. The block 0 contains the fixed parameters of the system inherent to the optimization problem, which are the weighing matrices $Q$ and $R$, the matrices $A$ and $B$ of the dynamic system, and the discount factor $\gamma$. The forgetting factor $\mu$, the parameter $\theta$, and the matrix $\Gamma$ are the necessary conditions for the RLS estimation problem.

**Algorithm 1 - RLS$_{\mu}$-ADHDP-DLQR**

<table>
<thead>
<tr>
<th>Block 0 – Initialization</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
</tr>
<tr>
<td>2.</td>
</tr>
<tr>
<td>3.</td>
</tr>
<tr>
<td>4.</td>
</tr>
<tr>
<td>5.</td>
</tr>
<tr>
<td>6.</td>
</tr>
<tr>
<td>7.</td>
</tr>
<tr>
<td>8.</td>
</tr>
<tr>
<td>9.</td>
</tr>
</tbody>
</table>

----------

<table>
<thead>
<tr>
<th>Block 1 – Environment Simulation</th>
</tr>
</thead>
<tbody>
<tr>
<td>10.</td>
</tr>
<tr>
<td>11.</td>
</tr>
<tr>
<td>12.</td>
</tr>
<tr>
<td>13.</td>
</tr>
<tr>
<td>14.</td>
</tr>
<tr>
<td>15.</td>
</tr>
<tr>
<td>16.</td>
</tr>
<tr>
<td>17.</td>
</tr>
<tr>
<td>18.</td>
</tr>
<tr>
<td>19.</td>
</tr>
<tr>
<td>20.</td>
</tr>
<tr>
<td>21.</td>
</tr>
</tbody>
</table>

----------

<table>
<thead>
<tr>
<th>Block 2 – Approximate Policy Evaluation</th>
</tr>
</thead>
<tbody>
<tr>
<td>22.</td>
</tr>
<tr>
<td>23.</td>
</tr>
<tr>
<td>24.</td>
</tr>
<tr>
<td>25.</td>
</tr>
<tr>
<td>26.</td>
</tr>
<tr>
<td>27.</td>
</tr>
<tr>
<td>28.</td>
</tr>
<tr>
<td>29.</td>
</tr>
<tr>
<td>30.</td>
</tr>
</tbody>
</table>

----------

<table>
<thead>
<tr>
<th>Block 3 – Policy Improvement (Feedback Optimal Gain K)</th>
</tr>
</thead>
<tbody>
<tr>
<td>31.</td>
</tr>
<tr>
<td>32.</td>
</tr>
</tbody>
</table>

----------

if $n_{\text{revit}} = 0$ then
4. Simulation Results

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 iteration technique. The typical parameters of configuration of the 3-DOF helicopter system are presented in Appendix A. The controller design via ADHDP is established aiming at maintaining the stabilization of the 3-DOF helicopter system in real-time, taking into account the iterative process configuration, as well 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 was carried out using the parameters given in Appendix A. The matrices of the linear continuous model are expressed as:

\[
A = \begin{bmatrix}
0 & 1 & 0 & 0 & 0 & 0 \\
0 & -0.753 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 \\
-0.098 & 0 & -1.192 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 \\
-1.257 & 0 & 0 & 0 & 0 & -0.457
\end{bmatrix}
\] (28)

And

\[
B = \begin{bmatrix}
0 & 0 \\
2.814 & -2.814 \\
0 & 0 \\
0.394 & 0.394 \\
0 & 0 \\
-0.035 & -0.035
\end{bmatrix}
\] (29)

The chosen output variables were the roll, elevation and travel angles \((y = [\phi \theta \psi]^T)\). Besides that, since there is no direct transmission between inputs and outputs of the plant, the C and D matrices were defined as

\[
C = \begin{bmatrix}
1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1
\end{bmatrix}
\]

and \(D = [0]\)

(30)

For many executed simulations, the influence of the forgetting factor \(\mu\) can be verified in the convergence process for the solution of the HJB-Riccati equation with different values of \(\mu\). In this approach, the algorithm presents a revitalization condition due to null-state problems that lead the matrix of regressors to a null rank [7]. So, it is necessary to perform the system revitalization after each interval \(n_{\text{revit}}\), which is given by the dimension of the regression vector \(\phi_k\).

For the implementation of the RLS-\(\mu\)-ADHDP-DLQR algorithm the initial conditions and system parameters are: an admissible initial policy \(K_0\); discount factor \(\gamma\); 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

\[
Q = [0.1 \ 0 \ 0 \ 0 \ 0; \ 0 \ 1 \ 0 \ 0 \ 0; \ 0 \ 0 \ 1 \ 0 \ 0; \ 0 \ 0 \ 0 \ 0 \ 0; \ ...; \ 0 \ 0 \ 0 \ 1 \ 0; \ 0 \ 0 \ 0 \ 0 \ 1]; R =
\]
The parameters that initialize the proposed RLS estimator are given by vector \( \theta_0 = 10^7 \times [1 0 0 0 0 0 0 1 0 0 0 0 1 0 0 0 0 \ldots 1 0 0 1 0 1] \); and matrix \( \Gamma_0 = 10 \times I_{36 \times 36} \), where \( I \) is an identity matrix of order \( n_\theta \). The matrices of the discretized system for the sampling interval \( T = 0.1 \) are obtained through the zero-orderhold (ZOH) method. For the simulations the following values of \( \mu \) are considered: 0.7640 and 0.9802.

### 4.1. Forgetting Factor \( \mu = 0.764 \)

The evolution of the iterative process of the \( Q \)-function estimation by the RLS\( _\mu \) – ADHDP – 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 components \( \theta_1, \theta_{13}, \theta_{16}, \theta_{27}, \theta_{25} \) and \( \theta_{35} \) of the parameter vector \( \theta \), 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-established flight condition, considering restrictions of the helicopter angles (which can be interpreted as obstacles that limit its maneuvering space), restrictions of control variables and external disturbance.

![Figure 3](image)

**Figure 3.** Evolution of the iterative process for the parameters \( s_{11}, s_{26}, s_{33}, s_{55}, s_{74} \) and \( s_{87} \) for 5000 cycle of iterations, with the forgetting factor \( \mu = 0.764 \) – RLS\( _\mu \) – ADHDP – 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 RLS\( _\mu \) – ADHDP – 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 4 indicate 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.
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 RLS-ADHDP-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 $S$ corresponds to the components $\theta_1$, $\theta_{13}$, $\theta_{16}$, $\theta_{27}$, $\theta_{25}$ and $\theta_{35}$ of the parameter vector $\theta$, 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 the factor $\mu = 0.764$ also happened to the factor $\mu = 0.9802$, the difference between the two factors is the accommodation time. One can observe throughout the simulations that for higher factors the convergence time tends to grow. For the forgetting factor $\mu = 0.9802$, the convergence was achieved after only 2500 iterations.
As for the previous factor, we have in Figures 7 and 8, respectively, the control effort added to the tracking noise and the trajectories of the system states. The chosen operating point remains the same, corresponding to the situation in which the helicopter finds itself still with elevation angle 14 degrees 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.
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 successful regarding 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 control process in real-time.

5. SIMULATION RESULTS

In the present design, the adaptive critic method, based on greedy policy iterations, was employed for 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 a process simulator for the ADHDP-DLQR design purpose. In the performance evaluation of the RLS$\mu$ – ADHDP – DLQR algorithm, one can verify with respect to the selection of the forgetting factor, 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 appropriate choice of the forgetting factor considerably improves the performance of the RLS$\mu$ – ADHDP – DLQR method. The use of $UDU^T$ factorization, as well as of the QR decomposition used in the 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 covariance matrix, causing a possible convergence loss of the systems under study.

ACKNOWLEDGEMENTS

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

Table 1. Parameter values

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\xi_1$</td>
<td>0.1117 N/V^2</td>
<td>$\xi_{11}$</td>
<td>1.0567 kg.m^2</td>
</tr>
<tr>
<td>$\xi_2$</td>
<td>0.0449 N/V</td>
<td>$\xi_{12}$</td>
<td>-0.2515 kg.m^2</td>
</tr>
<tr>
<td>$\xi_3$</td>
<td>-0.4843</td>
<td>$\xi_{13}$</td>
<td>0.5454 kg.m^2</td>
</tr>
<tr>
<td>$\xi_4$</td>
<td>0.1153</td>
<td>$\xi_{14}$</td>
<td>0.1258 kg.m^2</td>
</tr>
<tr>
<td>$\xi_5$</td>
<td>-1.0389 N/(m-kg)</td>
<td>$\xi_{15}$</td>
<td>0.5283 kg.m^2</td>
</tr>
<tr>
<td>$\xi_6$</td>
<td>-1.3170 N/(m-kg)</td>
<td>$\xi_{16}$</td>
<td>4.1832 (m-kg)^(-1)</td>
</tr>
<tr>
<td>$\xi_7$</td>
<td>0.0656 N/(m-kg.V^2)</td>
<td>$v_1$</td>
<td>0.107 N.m</td>
</tr>
<tr>
<td>$\xi_8$</td>
<td>0.0264 N/(m-kg.V)</td>
<td>$v_2$</td>
<td>0.18 N.s</td>
</tr>
<tr>
<td>$\xi_9$</td>
<td>-0.0718 N.m/V^2</td>
<td>$v_3$</td>
<td>0.47 N.m.s</td>
</tr>
<tr>
<td>$\xi_{10}$</td>
<td>-0.0289 N.m/V</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>

REFERENCES


