Adaptability of HDP-based DLQR Controllers Design for MIMO Dynamic Systems

Abstract - The increasing demand for high-performance control solutions has stimulated the development of optimal and robust controllers. For this purpose, an innovative method to evaluate the adaptability of the heuristic dynamic programming (HDP) method to approximate the solution of HJB-Riccati equation of the discrete linear quadratic regulator problem (DLQR) for online design of optimal controllers is presented in this article. The proposed method application is oriented to recursive least square (RLS) estimators that act on optimal approximations of HJB-Riccati equation solution, where adaptability of the estimators is evaluated in terms of the parameter variations on the plant model mappings concerning convergence speed and numerical stability properties of optimal solution. For two instances that can occur during operation time in real world systems (plants), the variations on the plant parameters are modelled to evaluate the performance of the RLS-HDP estimators to approximate the Hamilton-Jacobi-Bellman (HJB) equation solution. The instances are: a) sudden disturbances recovery that represents short time changes on the elements or devices of the plant, and b) an unscheduled operation point of the equipment must be established during the plant operation. The adaptability skills of the RLSµ-HDP-DLQR and RLSµ-UDU⁻¹-HDP-DLQR estimators are evaluated in a MIMO model of a dynamic system.

Keywords - Adaptability, Approximate Dynamic Programming, Discrete Linear Quadratic Regulator, Numerical Stability, Recursive Least Squares.

I. INTRODUCTION

The complexity of dynamic systems, due to performance requests of the design, demands high performance closed-loop controllers to execute the programmed tasks. An undesirable situation occurs when it is not possible to reach, simultaneously, all requests that are imposed by the designer during the controllers design. An alternative, to attend the design requests, is given in the optimal control theory that provides the means to develop solutions for dynamic systems that demand high performance controllers.

Optimal control solutions can be provided by the Bellman’s dynamic programming methods [1]–[3]. Nevertheless, the main drawback of this approach is that it is not suitable to real time applications because of the computational complexity associated with the Hamilton-Jacobi-Bellman (HJB) equation solution that is costly for large-scale multi-stage optimal control problems. In a general manner, the methods are offline and require the full knowledge of the dynamic system model. In fact, the model-based optimal solution is often difficult or impossible to be obtained due to inherent nonlinearities and/or time constant restrictions in real world systems (plant), as well as sensor costs and its installations.

An alternative to overcome that problem is the Adaptive Dynamic Programming (ADP) that provides real-time optimal approximate solutions of the HJB Equation and allows online tuning of controllers gains for systems with unknown dynamics. Different from the traditional dynamic programming, which require a “backward in time” procedure to solve the HJB equation, ADP solves the DP problems in a “forward in time” manner. This method is based on reinforcement learning to approximate optimal solution of a cost function that guarantees optimality over time. A survey of some research trends of ADP for future decades had been made in [4], [5]. In recent years, those areas have been recognized as promising for the realization of online designs of optimal control, [6]–[10].

In real-time implementations, the control plant may experience possible parametric variations that may change temporarily or permanently the system operating point, thus being necessary to assure that the controller compensates these disturbances. With that purpose in mind, a novel method to evaluate the adaptability of HDP-based DLQR controllers design for MIMO dynamic systems is presented in this paper, where adaptability of the RLS-HDP estimators of optimal cost function is evaluated with respect to parameter variations on the plant model. For two situations that can occur during operation time in real world systems (plant), the variations on the plant parameters are modelled to evaluate the performance of the RLS-HDP estimators to approximate the Hamilton-Jacobi-Bellman equation solution. The situations are: a) sudden disturbances recovery that represents short time changes on the elements or devices of the plant, and b) an unscheduled operation point of the equipment must be established during the plant operation. The adaptability skills of the RLSµ-HDP-DLQR and RLSµ-UDU⁻¹-HDP-DLQR estimators are evaluated in fourth order model of MIMO dynamic system.
This paper is organized into sections that present the development of a methodology and heuristic dynamic programming algorithms for the online design of DLQR controllers considering parametric variations on the plant. Initially, the theoretical foundations related to the HDP are presented in Section II, which comprise the concepts of Markov decision processes, dynamic programming, policy iteration schemes and temporal differences. These concepts are gathered as the online DLQR design framework. This section presents the development of standard structures to approximate the solution of HJB-Riccati equation via RLS-HDP estimator.

In Section III, the proposed method for online evaluation RLS-HDP-DLQR algorithms that are designed to perform the online tuning of DLQR controller gains is presented. The proposed performance evaluation method is based on disturbance or new values of the plant parameters. In Section IV, the results of experiments that were proposed to evaluate the adaptability of the RLS-HDP estimators are presented. The adaptability is investigated in terms of recovery to sudden disturbances and adaptability to new set points (steady state) during system operation. The performance of the RLS-DLQR-DLQR and RLS-DLQR-DLQR-DLQR estimators are evaluated regarding the convergence speed and numerical stability when there occurs deviations in the parameters or operational characteristics of the plant. The conclusions and comments to evaluate the performance of the optimal approximations of solution HJB-Riccati equation are presented in Section V.

II. ONLINE DLQR DESIGN FRAMEWORK

The standard heuristic dynamic programming (HDP) framework for online design of discrete linear quadratic regulator (DLQR) control systems is presented in this section. This framework is based on control policy strategies and Bellman Equation. The basic elements that are pointed to furnish the concepts of Bellman-DLQR problem formulation that is the junction of linear quadratic control theory and HDP method are presented. The solution method is also presented in its standard formulation.

A. Control Policy and Bellman Equation

The generalized formulation of the control policy and the Bellman equation which aims at showing the necessary elements for the development of a heuristic dynamic programming (HDP) scheme, such as the deterministic markovian decision process (MDP), dynamic programming and temporal differences that enables the problem characterization is presented here.

A deterministic MDP consists of a 5-tuple \( \mathcal{M} \equiv (X,U,f,r,\gamma) \), where \( X \) is the state space, \( U \) is the control action space, \( f : X \times U \rightarrow X \), \( f(x,u) = x_{k+1} \) is the deterministic state transition function, \( r : X \times U \rightarrow \mathbb{R} \); \( r(x,u) = r_i \) is the utility function that establishes the reward \( r_i \) of the transition from \( x_k \) to \( x_{k+1} \). This transition is guided (forced) by control action \( u_k \), and \( \gamma \) is the discount factor that is defined in \( 0 \leq \gamma \leq 1 \). For each state \( x_k \in X \) there is a subset \( U(x_k) \subseteq U \) of admissible actions, where \( u_k \) is an element of this subset.

The control policy is given by the mapping \( h : X \rightarrow U \) that produces an action \( u_k \) to be taken in time \( k \). For a given control policy \( h(x_k) = u_k \), the value-state function \( V^h : X \rightarrow \mathbb{R} \) is given by \( V^h(x_k) = \sum_{i=k}^{\infty} \gamma^{i-k} r(x_i,h(x_i)) \). For each \( x_k \in X \), \( V^h \) must agree with the equation that is given by

\[
V^h(x_k) = r(x_k,h(x_k)) + \gamma V^h(f(x_k,h(x_k))),
\]

that is called Bellman equation.

The MDP purpose is to establish a control or decision policy \( h^* \) that is optimal in the sense that it promotes the largest possible set of rewards, which satisfies the following inequality \( V^{h^*}(x_k) \geq V^h(x_k) \) for each \( x_k \in X \) and for all policies \( h_k \). According to Bellman-Optimality Principle, [11], the optimal value \( V^* \) is given by

\[
V^*(x_k) = \max_{h(\cdot)} \{ r(x_k,h(x_k)) + \gamma V^*(f(x_k,h(x_k))) \}
\]

and the optimal control policy \( h^* \) is given by

\[
h^*(x_k) = \arg\max\{ r(x_k,h(x_k)) + \gamma V^*(f(x_k,h(x_k))) \}.
\]

B. Bellman-DLQR Problem Formulation

The discrete linear quadratic regulator is characterized in the context of a deterministic MDP presented in Section II-A for HDP design purpose. The models \( f \) of the dynamic system and \( h \) of the control policy 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 are given by

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

and

\[
h(x_k) = -Kx_k
\]

where \( A \in \mathbb{R}^{n \times n} \), \( n \) is the system order, \( B \in \mathbb{R}^{n \times n} \), \( n \) is the number of the system inputs and \( K \in \mathbb{R}^{n \times n} \) is the gain matrix of the state feedback. It is assumed that \((A,B)\) is stabilizable, that is, there is a matrix \( K \) that guarantees that the closed loop system \( x_{k+1} = (A - BK)x_k \) is asymptotically stable.

The utility function \( r \) associated with the system \((4)-(5)\) has a quadratic form that is given by

\[
r(x_k,u_k) = x_k^TQx_k + u_k^TRu_k,
\]
where the weighting matrices \( Q \in \mathbb{R}^{n \times n} \geq 0 \) and \( R \in \mathbb{R}^{n \times n} \geq 0 \) are symmetric.

Replacing the parametrization of the utility function, Eq.(6), and decision (control) policy, Eq.(5), into the state-value function \( V^\gamma(x_k) \), one obtains the parameterized DLQR cost function. The DLQR control main purpose is to select a control policy \( K \) that minimizes a cost function that is given by

\[
V^K(x_k) = \sum_{i=k}^{\infty} \gamma^{i-k} x_i^T (Q + K^T R K) x_i, \quad \forall x_i \in X
\]

(7)

The optimal solution of DLQR, according to [12], admits the following quadratic form \( V^K(x_k) = x_k^T P x_k \), for some symmetric matrix \( P \in \mathbb{R}^{n \times n} \geq 0 \). Equation (7) and its quadratic form yield the same solutions. After algebraic manipulations with these equations, Bellman equation (1) for the DLQR is given by

\[
x_k^T P x_k = x_{k+1}^T \left[ Q x_{k+1} + u_k^T R u_k + \gamma(x_{k+1}^T P x_{k+1}) \right].
\]

(8)

Equation (8) in terms of the feedback gain of the parameterization of Eq.(5) and the dynamics of the closed loop system is expressed by

\[
x_k^T P x_k = x_k^T (Q + K^T R K) x_k + \gamma(A x_k + B u_k)^T P (A x_k + B u_k).
\]

(9)

Since Eq.(9) must be satisfied for all states \( x_k \), one has a linear equation in \( P \) that is given by

\[
\gamma(A - BK)^T P(A - BK) - P + Q + K^T R K = 0.
\]

(10)

If the gain \( K \) is fixed, Eq.(10) is known as Lyapunov equation. Given a stabilizable gain \( K \), the solution of this equation provides \( P = P^T > 0 \), such that \( V^K(x_k) = x_k^T P x_k \) is the cost value due to policy \( K \), writing the Bellman equation (8) as

\[
x_k^T P x_k = x_k^T \left[ Q x_k + u_k^T R u_k + \gamma(A x_k + B u_k)^T P (A x_k + B u_k) \right] = 0.
\]

(11)

Replacing this into Eq.(11), one obtains the discrete time HJB equation or the Bellman optimality equation for the DLQR parametrization

\[
\gamma(A^T P A) - P + Q - \gamma[A^T PB(R \gamma + B^T P B)^{-1} B^T P A] = 0.
\]

(12)

This equation is also known as discrete algebraic Riccati equation (DARE).

### C. RLS-HDP Approximation

Heuristic dynamic programming (HDP), which includes temporal difference methods, is established by a set of methodologies for the development of technics and procedures that estimate the value function.

The main idea behind the state-value RLS-HDP method is the estimation of the cost function \( V^K \) for a given policy \( K \) that only demands sampling from the instantaneous reward \( r \) and states, while the models of the environment and the utility function are needed to compute the cost function corresponding to the optimal policy. In this context, the RLS methods play the role of finding a parameter vector \( \theta \) for estimating \( V^K \). Consequently, the parametric structure of the linear approximation is given by

\[
V^K(x_k) = \phi^T(x_k) \theta
\]

(15)

and must agree with consistency condition that is given by

\[
V^K(x_k) = r(x_k, h_j(x_k)) + \gamma V^K(x_{h_j(x_k)}).
\]

(16)

The quadratic form of the cost function, is represented in terms of Kronecker product of states and in terms of matrix \( P \) vectorization that is the solution of the Hamilton-Jacobi-Bellman equation, Eq.(14). The optimal solution in its vectorized form is given by

\[
V^K_j(x_k) = x_k^T P_j x_k = \overline{x_k}^T vec(P_j),
\]

(17)

where \( \overline{x_k} \in \mathbb{R}^{n(n+1)/2} \) is a vector that is defined according to the Kronecker product given by

\[
\overline{x_k}^T = x_k^T \otimes x_k^T = \left[ x_{1,k}^2 \cdots x_{1,k}^n x_{2,k}^2 \cdots x_{n-1,k}^n x_{n,k}^{n-1} \right].
\]

(18)

where \( x_{l,k} \) is the \( l \)-th, \( l = 1,2,\ldots,n \), component of the state vector \( x \) at time \( k \), and \( vec(P_j) \in \mathbb{R}^{n(n+1)/2} \) results from the vectorization. This vector has the \( n \) diagonals elements of \( P_j \) and the \( n(n+1)/2 - n \) non-repeated sums \( p_{rs} + p_{sr} \). The elements of \( x_{l,k} \) and \( vec(P_j) \) are ordered so as to meet the quadratic form of Eq.(17).

Equation (16) is written in terms of the instantaneous reward and replacing the cost \( V^K_j(x_k) \) by the right-hand side of (17), one obtains a relation for the reward computation given by

\[
r(x_k, h_j(x_k)) = \overline{x_k}^T vec(P_j) - \gamma \overline{x_{k+1}}^T vec(P_j) = \phi^T \theta;
\]

(19)

where \( \phi_k = \overline{x_k} - \gamma \overline{x_{k+1}} \) and \( \theta = vec(P_j) \). It is noticed that the instantaneous reward \( r_{ij} \) is the desired value for estimation of parameter vector \( \theta \) in a least-squares sense. The algorithm for estimating \( \theta \) generates a vector sequence \( \{ \hat{\theta}(i) \}_{i=1}^N \).
given by
\[ \hat{\theta}_j(i) = \Phi_j^{-1}(i) \Theta_j(i), \]
with
\[ \Phi_j(i) = \sum_{m=1}^{N} \mu^{i-m} \phi_{k-i+m} \phi_{k-i+m}^T \]
and
\[ \Theta_j(i) = \sum_{m=1}^{N} \mu^{i-m} \phi_{k-i+m} \Gamma_{k-i+m}, \]
where \( \mu \) is the forgetting factor and \( N \) is the number of sampling data.

By simple algebraic manipulations in the last three equations and by applying the matrix inversion lemma, the recursive estimation of \( \theta \) is splined in three equations that are the gain matrix, the parameter vector \( \theta \) and the inverse of the matrix \( \Phi \), which is denoted by \( \Gamma \). The RLS gain in step \( i \) is given by
\[ L_j(i) = \Gamma_j(i) \mu_k = \frac{\Gamma_j(i-1) \phi_k}{\mu_k + \phi_k^T \Gamma_j(i-1) \phi_k}, \]
where \( j \) is the index for policy update, \( i \) is the index of the current recurrence of the RLS estimator, \( k \) is the discrete time. The parameter estimation in time \( i \) is given by
\[ \hat{\theta}_j(i) = \hat{\theta}_j(i-1) + L_j(i) \left( r_k - \phi_k^T \hat{\theta}_j(i-1) \right), \]
where \( \hat{\theta}_j(i) \) is the \( i \)-th estimate of \( \theta \). The covariance matrix in step \( i \) is given by
\[ \Gamma_j(i) = \mu^{-1} \left[ \Gamma_j(i-1) - L_j(i) \phi_k^T \Gamma_j(i-1) \right], \]
where \( \Gamma_0 = \delta I \) for some positive constant \( \delta \), \( \Gamma_{i+1}(0) = \Gamma_i \) and \( \Gamma(0) = \Gamma_0 \).

### III. ONLINE ALGORITHMS EVALUATION METHOD

This section presents the proposed method to evaluate the adaptability of RLS-HDP-DLQR control system design for parametric disturbances on the plant. The proposed method has two main components: a) the plant parametric disturbance model that is based on methods developed by [13], b) the RLS-HDP-DLQR algorithms that are based on methods developed by [14].

#### A. Plant Parametric Disturbance Procedure

This section presents the procedure related to parametric disturbances on the plant to evaluate the algorithms performance. These algorithms were developed for the online DLQR control design via HDP. One of questions of interest in the analysis of the results is to verify the adaptability of the RLS-HDP-DLQR and RLS-HDP-DLQR estimators in face of the parametric variations on the plant from a previous stable condition.

Consider a dynamic system described by the following state equation
\[ x(t) = A(p(t)) x(t) + B(p(t)) u(t) \]
\[ y(t) = C(p(t)) x(t), \]
where \( A \in \mathbb{R}^{n \times n}, x \in \mathbb{R}^n \) is the state vector, \( B \in \mathbb{R}^{n \times m} \), \( u \in \mathbb{R}^m \) is the control input vector, and \( p \) is a time-varying and exogenous parameter (strictly independent of the state \( x \)). The parameter \( p \) can, in general, be unknown a priori and it can be measured or estimated on the system operation. Typical guest on \( p \) are primarily the limits on its magnitude and ranging rate.

The model of Eq. (23) was inspired in the works by Shamma, [13], [15], [16], who proposed a framework of Linear Parameter Varying models for analysis and design of gain-scheduling control systems. Such systems consist of a control design approach which builds a non-linear controller for a non-linear plant from a family of linear controllers indexed to a exogenous parameter. These controllers are combined in real time according to available online measurements.

In order to evaluate the behavior of the RLS-HDP-DLQR and RLS-HDP-DLQR estimators with respect to changes governed by the parameter \( p(t) \) in the model (23), the following trajectories for \( p(t) \) are proposed:
\[ p_1(t) = \begin{cases} 0 & \text{if } t_0 \leq t < t_v \\ \alpha e^{\beta (t-t_v)} & \text{if } t \geq t_v \end{cases} \]
\[ p_2(t) = \begin{cases} 0 & \text{if } t_0 \leq t < t_v \\ \alpha - \alpha e^{\beta (t-t_v)} & \text{if } t \geq t_v \end{cases} \]
where \( \alpha > 0 \) and \( \beta > 0 \) are constants. A motivation to consider parametric variations governed by exponentsials such as \( p_1 \) and \( p_2 \) comes from the fact that those may represent real world situations where the systems are subject to more smooth changes at their operating point. However, it is observed next some limit cases of \( p_1 \) and \( p_2 \) which may represent more abrupt variations of the operating point of the system which include disturbances governed by a step and a quite fast duration pulse. The limit cases of the functions \( p_1 \) and \( p_2 \) are given by

a) \( p_1 \) when \( \beta \rightarrow \infty \): pulse of duration \( k \).
b) \( p_2 \) when \( \beta \rightarrow \infty \):
\[ p_2^\infty(t) = \begin{cases} \alpha & \text{if } t \geq t_v \\ 0 & \text{otherwise} \end{cases} \]
c) \( p_1 \) when \( \beta \rightarrow 0^+ \): the function \( p_1 \) approaches \( p_2^\infty \).
d) \( p_2 \) when \( \beta \rightarrow 0^+ \): there is no parametric variation on the plant.
B. RLS-HDP Algorithm for DLQR

The standard RLS-HDP-DLQR algorithm has two main loops that are the iterative processes of the HDP and the RLS for determining the policy decision (control) based on reinforcement learning methods. In this situation the customized algorithm is made up of the following main steps:

A full view of Algorithm 1 (RLS-HDP-DLQR) and whole description of its functioning can be seen in [14].

C. UDU^T RLS-HDP-DLQR Algorithm

Results related to RLS-HDP-DLQR estimators with UDU^T factorization are obtained with the implementation of Eqs.(27)-(38) for computing the update of \( \tilde{\theta}_j^i \). These equations are grouped to the previous algorithm procedures to set up Algorithm 2, called UDU^T RLS-HDP-DLQR Algorithm.
Block 3 realizes the approximate policy evaluation based on the following equations of the RLS estimator with $UDU^T$ factorization.

$$
\varepsilon_j(i) = r_k - \phi_k^T \theta_j(i - 1) \tag{27}
$$

$$
e = U(2g - 1) \phi_k \tag{28}
$$

$$
g = D(i-1)e \tag{29}
$$

$$
\beta_0 = \mu \tag{30}
$$

For $l$ from 1 to $n_d$ do

$$
\beta_l = \beta_{l-1} + e_l g_l \tag{31}
$$

$$
d_l(i) = d_l(i-1) \frac{\beta_{l-1}}{\beta_l} \mu \tag{32}
$$

$$
\kappa = g_l \tag{33}
$$

$$
\tau_{\kappa} = -e_l \tag{34}
$$

If $l \neq 1$, for $m$ from 1 to $l-1$, do

$$
u(i) = u(i-1) + \tau \kappa_m \tag{35}
$$

$$
\kappa_{m,\text{new}} = \kappa_{m,\text{old}} + u_m(i-1) \kappa_l \tag{36}
$$

Gain vector updating

$$
L_j(i) = \begin{bmatrix} \kappa_1 & \kappa_2 & \ldots & \kappa_{n_d} \end{bmatrix} \tag{37}
$$

Parameter vector updating

$$
\tilde{\theta}_j(i) = \tilde{\theta}_j(i - 1) + L_j(i) \varepsilon_j(i) \tag{38}
$$

All explanation about Algorithm 2 ($UDU^T$ RLS-HDP-DLQR) in special Block 3 can be found in [14]

### IV. COMPUTATIONAL EXPERIMENTS

The setup of computational experiments constitute the first part of the procedure for the performance evaluation of the algorithms for online controllers design via HDP. In the second part of the procedure, comparisons are performed to evaluate the accuracy of solutions of the controllers (policies). The reference policy is established by offline Schur solution of the HJB equation, which may be considered as exact solution. It is used to show that the approximate HJB solution has the ability not only to reach a solution sufficiently close to exact solution but also to achieve the exact solution.

**A. Setup of the Iterative Process**

The setup of the iterative process consists in establishing the better parameters for solving a given application, having as reference the designer’s empiricism or the relationships among system variables related to the methods. The parameters that play a relevant role in the method convergence are: system order $n$ ($n = 4$), sample interval $T_{\text{amort}}$ ($T_{\text{amort}} = 0.1s$) and forgetting factor $\mu$ ($\mu = 0.92$).

1) **Dynamic System Matrices:** The dynamic system matrices represent the behavior of a fourth-order electric circuit that is given in reference [17]. The matrices of the dynamic system for the continuous time state space description are given by

$$
A_{TC} = \begin{bmatrix} -0.001 & 0 & 10 & 0 \\ 0 & -0.001 & 0 & -10 \\ -0.1 & 0 & -10 & -10 \\ 0 & 0.1 & -10 & -10 \end{bmatrix} \tag{39}
$$

and

$$
B_{TC} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0.1 & 0 \\ 0 & 0.1 \end{bmatrix} \tag{40}
$$

The matrices of the discretized system for the sampling interval $T_{\text{amort}} = 0.1$ are obtained by the impulse-invariant method.

2) **Initial Conditions for RLS:** The initial parameter vector $\theta_0$ of the Riccati/Lyapunov equation estimated by the RLS method is given by $\theta = [\theta_1, \theta_2, \ldots, \theta_{n_d}]^T$, where $\theta_i = 0$ for $i = 1, \ldots, 10$. The startup corresponds to the fourth-order dynamic system model.

The iterative process startup of the approximate policy iteration requires the selection of an admissible initial policy $K_0$. For such a case, the initial matrix of the Riccati/Lyapunov equation for RLS is given by

$$
P_0 = \rho L_k \kappa_0 \tag{41}
$$
where \( I_{nx1} \) is the identity matrix and \( \rho \) is a real scalar (\( \rho = 10 \)). This matrix is associated with the parameters of the RLS approximation for the fourth-order dynamic system, according to the formation law of Eq.(17).

| Case | \( \alpha \) | \( \beta \) | \( \kappa = \frac{\alpha}{\beta} \) | HDP | RLS
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.25</td>
<td>0+</td>
<td>0.92</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>0+</td>
<td>0.92</td>
<td>-</td>
<td>†</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>0.01</td>
<td>0.92</td>
<td>-</td>
<td>†</td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>0.1</td>
<td>0.92</td>
<td>-</td>
<td>†</td>
</tr>
<tr>
<td>5</td>
<td>1</td>
<td>1</td>
<td>0.92</td>
<td>-</td>
<td>†</td>
</tr>
<tr>
<td>6</td>
<td>1</td>
<td>10</td>
<td>0.92</td>
<td>-</td>
<td>†</td>
</tr>
<tr>
<td>7</td>
<td>1</td>
<td>100</td>
<td>0.92</td>
<td>-</td>
<td>†</td>
</tr>
<tr>
<td>8</td>
<td>10</td>
<td>0.001</td>
<td>0.92</td>
<td>-</td>
<td>†</td>
</tr>
<tr>
<td>9</td>
<td>10</td>
<td>0.1</td>
<td>0.92</td>
<td>-</td>
<td>†</td>
</tr>
<tr>
<td>10</td>
<td>10</td>
<td>1</td>
<td>0.92</td>
<td>-</td>
<td>†</td>
</tr>
<tr>
<td>11</td>
<td>10</td>
<td>10</td>
<td>0.92</td>
<td>-</td>
<td>†</td>
</tr>
<tr>
<td>12</td>
<td>10</td>
<td>100</td>
<td>0.92</td>
<td>-</td>
<td>†</td>
</tr>
<tr>
<td>13</td>
<td>10</td>
<td>1013</td>
<td>0.92</td>
<td>-</td>
<td>†</td>
</tr>
<tr>
<td>14</td>
<td>10</td>
<td>∞</td>
<td>0.92</td>
<td>-</td>
<td>†</td>
</tr>
</tbody>
</table>

### Table I Parametric disturbance Cp1

| Case | \( A \) | \( B \) | \( \alpha \) | \( \kappa = \frac{\alpha}{\beta} \) | HDP | RLS
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>0.001</td>
<td>0.92</td>
<td>-</td>
<td>†</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>0.1</td>
<td>0.92</td>
<td>-</td>
<td>†</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>1</td>
<td>0.92</td>
<td>-</td>
<td>†</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>10</td>
<td>0.92</td>
<td>-</td>
<td>†</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>1</td>
<td>100</td>
<td>0.92</td>
<td>-</td>
<td>†</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>1</td>
<td>1000</td>
<td>0.92</td>
<td>-</td>
<td>†</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>1</td>
<td>0.001</td>
<td>0.92</td>
<td>-</td>
<td>†</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>1</td>
<td>0.1</td>
<td>0.92</td>
<td>-</td>
<td>†</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>1</td>
<td>1</td>
<td>0.92</td>
<td>-</td>
<td>†</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>1</td>
<td>10</td>
<td>0.92</td>
<td>-</td>
<td>†</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>1</td>
<td>100</td>
<td>0.92</td>
<td>-</td>
<td>†</td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>1</td>
<td>1013</td>
<td>0.92</td>
<td>-</td>
<td>†</td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>1</td>
<td>∞</td>
<td>0.92</td>
<td>-</td>
<td>†</td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>1.5</td>
<td>∞</td>
<td>0.92</td>
<td>-</td>
<td>†</td>
<td></td>
</tr>
</tbody>
</table>

### Table II Parametric disturbance Cp2

B. Parametric Disturbance Experiments

Tables I and II, present the parametric variations of the type \( p_1 \) and \( p_2 \), respectively, for different values of \( \alpha \) and \( \beta \). These variations were applied in the parameters of the plant of the electrical circuit system, which follows the model of Eq.(23), details of MIMO plant can be seen in [17]. The symbol † indicates if there occurs the non-adaptability of the RLS-UDU-HDP-DLQR estimators for disturbances like \( P_1^k \), or equivalently of the type \( p_1^k \). It is observed that the learning process of the RLS-HDP-DLQR estimator becomes slower and with the presence of greater peaks with respect to RLS-UDU-HDP-DLQR estimator, as can be verified in Figures 1-3 which illustrate the evolution of the parameters of the RLS-UDU-HDP-DLQR estimators for a cycle of 5000 iterations, with the constant \( \alpha \) equal to 0.25. In other experiments carried out, it was observed that the RLS-UDU-HDP-DLQR estimators present adaptability for a variation range around (0,1.5) of the constant \( \alpha \), whereas the RLS-HDP-DLQR estimators present adaptability for a range of values \( \alpha \) in (0,0.25).

![Figure 1](image1.png)

**Fig. 1.** Parametric disturbance \( p_2 \) with \( \alpha = 0.25 \) and \( \beta = \infty \) - Evolution of the iterative process for the \( p_1 \), parameters to a cycle of 5000 iterations, with forgetting factor \( \mu = 0.92 \) - RLS-UDU-HDP-DLQR algorithm.

![Figure 2](image2.png)

**Fig. 2.** Zoom in parametric variation of type \( p_2 \) in the iterative process of the parameter \( p_{11} \) on Figure 1.
Fig. 3. Parametric disturbance $p_2$ with $\alpha = 0.25$ and $\beta = \infty$ - Evolution of the iterative process for the $p_n$ parameters to a cycle of 5000 iterations, with forgetting factor $\mu = 0.92$ - RLS$_\mu$-HDP-DLQR algorithm.

With respect to the disturbances of the type $p_2$, for larger values of $\beta$, a behavior similar to that caused by disturbance of the type $P_{2}^\infty$ was verified. Figures 4 and 5 represent the behavior of the RLS$_\mu$-UDU$^T$-HDP-DLQR estimator along the learning process in face of the disturbances of the type $p_2$ for values of $\beta$ equal to 10, with $\alpha = 0.25$. A similarity between the behaviors of each of the estimators for the cases $\alpha = 0.25$, $\beta > 10^{13}$ and $\alpha = 0.25$, $\beta = \infty$ was verified.

As to the disturbances of the type $p_2$, for larger values of $\beta$, a behavior similar to that caused by disturbance of the type $P_{2}^\infty$ was verified. Figures 4 and 5 represent the behavior of the RLS$_\mu$-UDU$^T$-HDP-DLQR estimator along the learning process in face of the disturbances of the type $p_2$ for values of $\beta$ equal to 10, with $\alpha = 0.25$. A similarity between the behaviors of each of the estimators for the cases $\alpha = 0.25$, $\beta > 10^{13}$ and $\alpha = 0.25$, $\beta = \infty$ was verified.

Fig. 4. Parametric disturbance $p_2$ with $\alpha = 0.25$ and $\beta = 10$ - Evolution of the iterative process for the $p_n$ parameters to a cycle of 5000 iterations, with forgetting factor $\mu = 0.92$ - RLS$_\mu$-UDU$^T$ - HDP-DLQR algorithm.

Fig. 5. Zoom in parametric variation of type $p_2$ in the iterative process of the parameter $p_{11}$ on Figure 4.
Fig. 6. Parametric disturbance $p_1$ with $\alpha = 10$ and $\beta = 10$ - Evolution of the iterative process for the $p_i$ parameters to a cycle of 5000 iterations, with forgetting factor $\mu = 0.92$ - RLS$_\mu$-UDU$^T$-HDP-DLQR algorithm.

Fig. 7. Zoom in parametric variation of type $p_1$ in the iterative process of the parameter $p_{11}$ on Figure 6.

It can be observed that the types of disturbances $p_1$ and $p_2$ that have more influence on the behaviors of the RLS$_\mu$-HDP-DLQR and RLS$_\mu$-UDU$^T$-HDP-DLQR estimators correspond to cases where $\beta = 0.001$, as can be seen from Figures 8-10 because the convergence process takes longer regarding the convergence of cases highlighted earlier.

Fig. 8. Parametric disturbance $p_2$ with $\alpha = 1$ and $\beta = 0.001$ - Evolution of the iterative process for the $p_i$ parameters to a cycle of 15000 iterations, with forgetting factor $\mu = 0.92$ - RLS$_\mu$-UDU$^T$-HDP-DLQR algorithm.

Fig. 9. Zoom in parametric variation of type $p_2$ in the iterative process of the parameter $p_{11}$ on Figure 8.

Fig. 10. Parametric disturbance $p_2$ with $\alpha = 1$ and $\beta = 0.001$ - Evolution of the iterative process for the $p_i$ parameters to a cycle of 12000 iterations, with forgetting factor $\mu = 0.92$ - RLS$_\mu$-HDP-DLQR algorithm.
In general, the RLS\(_\mu\)-UDU\(^T\)-HDP-DLQR estimators yielded better results when compared to the RLS\(_\mu\)-HDP-DLQR estimators, presenting a faster adaptability process with respect to variations of the type \(p_1\) and \(p_2\). Whereas the RLS\(_\mu\)-HDP-DLQR estimators showed less adaptable to variations on the plant in most cases considered in the present experiments, especially for variations of the type \(p_1\) where no adaptability took place for none of the situations, except for the \(p_1^0\) case, \(\alpha \in (0, 0.25]\). Nevertheless, an interesting case is illustrated by Figures 11-13 for the disturbance case \(p_2\), \(\alpha = 1, \beta = 0.1\), where the RLS\(_\mu\)-HDP-DLQR estimators managed to adapt more quickly to plant variations.

Fig. 11. Parametric disturbance \(p_2\) with \(\alpha=1\) and \(\beta=0.1\) - Evolution of iterative process of the iterative process for the \(p_{ii}\) parameters to a cycle of 6000 iterations, with the parameter \(p_{11}\) forgetting factor \(\mu = 0.92\) - RLS\(_\mu\)-UDU\(^T\)-HDP-DLQR algorithm.

Fig. 12. Zoom in parametric variation of type \(p_2\) in the on Figure 11.

Fig. 13. Parametric disturbance \(p_2\) with \(\alpha=1\) and \(\beta=0.1\) - Evolution of the iterative process for the \(p_{ii}\) parameters to a cycle of 6000 iterations, with forgetting factor \(\mu = 0.92\) - RLS\(_\mu\)-HDP-DLQR algorithm.

V. CONCLUSION

A novel method that aims at evaluating the adaptability of HDP-based DLQR controllers design for MIMO dynamic systems was presented. The behavior of the RLS\(_\mu\)-HDP-DLQR and RLS\(_\mu\)-UDU\(^T\)-HDP-DLQR estimators was evaluated in terms of the proposed model for parameter variations on the plant dynamics.

An instance of the model simulated a short time disturbance on the plant, being it a RLC circuit that can be represented as an oscillation disturbance of the power supply. In this simulation it was noticed that the system was sensitive to that disturbance but returned to its normal operating point with a good speed.

The second situation represented a change in the operating point on the plant. In the real world it would be an increase or decrease of the load on the circuit, so the system had to adapt to a new plant having to find a new control gain. In this simulation, it was observed that the more drastic the change to this new point, the longer and with more oscillations was the system response, but it attained this new solution.

In general, the RLS\(_\mu\)-UDU\(^T\)-HDP-DLQR estimators yielded better results when compared to the estimators RLS\(_\mu\)-HDP-DLQR, with a faster adaptability process with respect to model of parametric variations on the plant presented in this paper. It was shown that large performance improvements can be achieved by incorporating \(UDU^T\) factorization into the process of matrix \(\Phi\) inversion of the RLS-HDP estimator.
REFERENCES


