

Control of Nonlinear Stochastic Systems: Model-Free Controllers versus Linear Quadratic Regulators

Vural Aksakalli and Daniel Ursu

Abstract— Classical linear controllers are widely used in the control of nonlinear stochastic systems and thus there is concern about the ability of the controller to adequately regulate the system. An alternative approach to cope with such systems is to avoid the need to build the traditional “open-loop” model for the system. Through the avoidance of model, controllers can be built for arbitrarily complex nonlinear systems via neural-networks trained by simultaneous perturbation stochastic approximation so that only the output error (between the plant and target outputs) is needed. In this paper, we discuss basic characteristics and limitations of both approaches and formally analyze this comparison in the case of linear quadratic regulators. The comparison is illustrated numerically on a simulated nonstationary multiple input, multiple output wastewater treatment system with stochastic effects.

I. INTRODUCTION

MODERN control engineering is expanding rapidly to fill the needs in complex and challenging systems for regulation and control. Such modern systems go well beyond the traditional electrical, mechanical, and aerospace systems that have been at the heart of control systems research for many years. Included in the kinds of modern systems for which control is needed are, to name a few, communications and transportation networks [1], biomedical systems (e.g., automated surgery and drug delivery [2]), and the control of financial markets [3]. Such modern systems do not typically lend themselves to easy representation via linear differential equations. Hence, the majority of the techniques that have been developed over the years to control linear systems may be inappropriate for coping with the control of many modern systems. Furthermore, despite the considerable efforts of many researchers and practitioners over many years, formal control techniques for most real-world nonlinear systems are unavailable [4]. Simply put, closed-form (or other “easy”) solutions to nonlinear problems are almost never available and hence linear methods are generally used.

Spall and Cristion [5] make a significant advance in coping with nonlinear, stochastic systems by using neural network based controllers trained via simultaneous

perturbation stochastic approximation (SPSA) so that the need to build the traditional open-loop model is avoided. The approach presented therein is based on using the output error of the system to directly train the NN controller without the need for a separate model (NN or other type) for the unknown process equations. Since it is assumed that the system dynamics are unknown, determining the gradient of the loss function in typical back-propagation type weight estimation algorithms is not feasible. To implement such a direct adaptive control, the authors propose simultaneous perturbation stochastic approximation for estimating the NN connection weights while the system is being controlled. In a related work, the authors demonstrate how such a model-free controller can be efficiently utilized to control a challenging nonlinear multiple input, multiple output (MIMO) stochastic wastewater treatment problem [6]. The model-free approach, although relatively new, has already been applied successfully in many real-world control problems [8,9,10,11]. However, a comparison with classical linear methods, theoretical or numerical, has not yet been conducted.

Our goal in this paper is to formally analyze such a comparison in the case of linear quadratic regulators (LQR) and illustrate this comparison on an empirical basis in a challenging nonlinear control problem encountered in wastewater treatment systems. Our purpose is to provide some insight into the value of the model-free method and motivate further research in this direction.

The rest of this paper is organized as follows. Section II outlines model-free and classical controllers and briefly discusses their basic characteristics and limitations. Section III presents LQR and compares it to the model-free controller. Section IV describes the wastewater treatment problem and discusses implementations of the model-free controller and LQR for the problem, followed by a comparison of the two approaches. Section V summarizes our findings and relates them to those of Spall and Cristion [5,6] and Dochain and Bastin [7]. Directions for future research are discussed in Section VI.

II. MODEL-FREE & CLASSICAL CONTROLLERS: BASIC CHARACTERISTICS AND LIMITATIONS

A. Model-Free Controllers

We consider a discrete-time state vector given by

$$x_{k+1} = \phi_k(x_k, u_k, w_k), \quad (2.1)$$

where ϕ_k is a nonlinear, yet *unknown* function governing

Authors would like to thank Dr. James C. Spall of The Johns Hopkins University's Applied Physics Laboratory for his constructive comments and suggestions.

V. Aksakalli is with the Department of Applied Mathematics and Statistics, The Johns Hopkins University, Baltimore, MD 21218 USA (e-mail: ala@jhu.edu).

D. Ursu is with the Department of Mechanical Engineering, The Johns Hopkins University, Baltimore, MD 21218 USA (e-mail: dursu1@jhu.edu).

the system dynamics, u_k is the control input applied to the system at time k , and w_k is a random noise vector. Our focus will be on the case of direct measurements as in [5] and [6]. The goal here is to determine the control vectors $\{u_k\}$ such that the state values $\{x_k\}$ are as close as possible to a set of target vectors $\{t_k\}$. The information fed into the NN controller consists of the next target vector, M most recent state values, and N most recent controls. The output of the NN is then the value of the control u_k . Associated with this NN is a vector of connection weights $\theta_k \in \mathbf{R}^p$ that will be estimated. Our goal is to find θ_k^* that minimizes some loss function $L(\theta)$ measuring system performance. We will use the one-step-ahead quadratic tracking error below as the performance criterion:

$$L_k(\theta_k) = E \left[(x_{k+1} - t_{k+1})^T W_k (x_{k+1} - t_{k+1}) + u_k^T Z_k u_k \right] \quad (2.2)$$

where W_k and Z_k are positive semi-definite matrices specifying the relative weights on the deviations from the target values and the cost of large control values.

To find the optimal values of θ_k , the model-free controller uses a stochastic approximation of the form

$$\hat{\theta}_k = \hat{\theta}_{k-1} - a_k \hat{g}_k(\hat{\theta}_{k-1}), \quad (2.3)$$

where $\hat{g}_k(\hat{\theta}_{k-1})$ is the simultaneous perturbation approximation to $g_k(\hat{\theta}_{k-1}) = \partial L_k / \partial \theta_k$. The reader is referred to [1] for an in-depth discussion of the SPSA-based NN controllers. We now briefly discuss basic characteristics and limitations of this approach.

1. The use of “model-free” is to be taken literally in the sense that no hidden or implicit modeling is required, which eliminates the *system characterization and identification* processes, and thus the need to allocate time and resources to determine an adequate model of the underlying system and evaluate its validity.
2. Three major advantages of the model-free controllers are that they (i) tend to better handle changes in the underlying system as they are not tied to a prior model, (ii) require no open-loop training data, and (iii) tend to be more robust in the case of widely varying control inputs.
3. The model-free approach is appropriate for many practical systems, yet it is generally inappropriate for systems where a reliable model of the system can be determined.
4. The model-free controller requires that the system under consideration be approximately stationary while an individual SPSA approximation is performed (the system dynamics can be nonstationary over longer time periods, however). A further restriction (which is typical of controllers relying on imperfect prior system knowledge) is that the system be able to tolerate suboptimal controls as the learning process takes place.
5. Success of the model-free approach in any particular

application depends on the choice of the NN structure and SPSA implementation methodology.

B. Classical Controllers

Discrete-time MIMO linear time-invariant systems are defined as $x_{k+1} = Ax_k + Bu_k$ (assuming direct state measurements); where A and B are matrices determined via a system identification process. Fundamental characteristics and limitations of classical linear controllers are briefly discussed below.

1. These controllers are widely used in practice due to their simplicity and availability of corresponding software tools and commercial products.
2. Given a nonlinear system, classical controllers can be used only on a “linearized” version of the system, giving good results at an equilibrium point about which the system behavior is approximately linear. However, this assumption of linearity is usually violated to a certain degree in many of today’s complex control systems.
3. Such controllers show poor and/or inadequate performance when process and/or measurement noise is present and in the case where the system varies in time.

III. LQR VERSUS MODEL-FREE CONTROLLERS: A FORMAL COMPARISON

A. Linear Quadratic Regulators

Similar to model-free controllers, linear quadratic regulators (LQR) involve minimization of a loss function measuring the difference between the system’s outputs and target outputs. The performance criterion used in LQR is the following quadratic loss function:

$$J = \sum_{k=1}^{K-1} \begin{bmatrix} e_k^T & v_k^T \end{bmatrix} Q \begin{bmatrix} e_k \\ v_k \end{bmatrix} + u_k^T R u_k \quad (3.1)$$

Above, e_k is the control error (i.e., $e_k = x_k - t_k$), v_k is the cumulative error ($v_k = \sum_{i=1}^{k-1} e_i$), K is the number of iterations,

and u_k is the control input. The goal is to determine the control sequence $\{u_k\}$ such that J is minimized. The matrices Q and R reflect the relative weights of control errors and the control gains. The above criterion poses interest to us because of its similarity to (2.2); the performance criterion of the model-free controller. The implementation of this algorithm is as follows [13, Chap. 8]: the loss function is first rewritten as

$$J = \sum_{k=0}^{K-1} [x_k^T P x_k] \quad (3.2)$$

where P is defined as the optimal steady-state matrix. For a linear system described by $x_{k+1} = Ax_k + Bu_k$, P is given by

$$\begin{aligned} P &= Q + A^T P (I + BR^T B)^{-1} A \\ &= Q + A^T (P^{-1} + BR^{-1} B^T)^{-1} A \end{aligned}$$

The above Riccati equation is solved iteratively until \mathbf{P} no longer changes values. The above expression further shows that \mathbf{P} is solely dependent on the state-space matrices \mathbf{A} and \mathbf{B} , and the matrices \mathbf{Q} and \mathbf{R} associated with the loss function. The steady-state gain matrix \mathbf{K} can then be written in terms of \mathbf{P} as:

$$\mathbf{K} = (\mathbf{R} + \mathbf{B}^T \mathbf{P} \mathbf{B})^{-1} \mathbf{B}^T \mathbf{P} \mathbf{A} \quad (3.3)$$

The matrix \mathbf{K} optimizes the actual input, so the control law that minimizes (3.1) becomes:

$$\mathbf{u}_k = -\mathbf{K} \mathbf{x}_k \quad (3.4)$$

Once fed back into the original control problem, the control system can be stated as follows using (3.3) and (3.4):

$$\mathbf{x}_{k+1} = [\mathbf{A} - \mathbf{B}(\mathbf{R} + \mathbf{B}^T \mathbf{P} \mathbf{B})^{-1} \mathbf{B}^T \mathbf{P} \mathbf{A}] \mathbf{x}_k \quad (3.5)$$

B. LQR versus Model-Free Controllers

The LQR differs from the model-free controller in the sense that the former assumes a modeled control process whereas the latter does not. However, contrasting (3.1) to (2.2), it can be seen that both controllers attempt to minimize similar loss functions in an iterative manner. We now formalize this connection between the two controllers.

The objective in the model-free framework is to determine the control vector \mathbf{u}_k that minimizes the one-step-ahead tracking error where $\mathbf{x}_{k+1} = \phi_k(\mathbf{x}_k, \mathbf{u}_k, \mathbf{w}_k)$. Assuming that the model-free framework uses constant gain and error matrices as in (2.2), we have

$$\mathbf{u}_k^{mf} = \mathbf{u}_k(\boldsymbol{\theta}_k^*; \mathbf{x}_k, \mathbf{x}_{k-1}, \dots, \mathbf{x}_{k-M+1}; \mathbf{u}_{k-1}, \mathbf{u}_{k-2}, \dots, \mathbf{u}_{k-N+1}; \mathbf{t}_{k+1}),$$

where

$$\boldsymbol{\theta}_k^* = \arg \min_{\boldsymbol{\theta}_k} \{ (\mathbf{x}_{k+1} - \mathbf{t}_{k+1})^T \mathbf{W} (\mathbf{x}_{k+1} - \mathbf{t}_{k+1}) + \mathbf{u}_k^T \mathbf{Z} \mathbf{u}_k \}.$$

With a slight abuse of notation, we shall write

$$\mathbf{u}_k^{mf} = \arg \min_{\mathbf{u}_k} \{ (\mathbf{x}_{k+1} - \mathbf{t}_{k+1})^T \mathbf{W} (\mathbf{x}_{k+1} - \mathbf{t}_{k+1}) + \mathbf{u}_k^T \mathbf{Z} \mathbf{u}_k \}. \quad (3.6)$$

Let the time-invariant linearization of ϕ be $\hat{\mathbf{x}}_{k+1} = \mathbf{A} \mathbf{x}_k + \mathbf{B} \mathbf{u}_k$ where \mathbf{A} and \mathbf{B} represent the linear system analogous to that of the LQR control. Define linear approximation residuals as

$$\begin{aligned} r_{k+1} &= \mathbf{x}_{k+1} - \hat{\mathbf{x}}_{k+1} \\ &= \phi_k(\mathbf{x}_k, \mathbf{u}_k, \mathbf{w}_k) - (\mathbf{A} \mathbf{x}_k + \mathbf{B} \mathbf{u}_k) \end{aligned} \quad (3.7)$$

The control error \mathbf{e}_{k+1} can now be written as

$$\begin{aligned} \mathbf{e}_{k+1} &= \mathbf{x}_{k+1} - \mathbf{t}_{k+1} \\ &= \hat{\mathbf{x}}_{k+1} + r_{k+1} - \mathbf{t}_{k+1} \end{aligned}$$

Let $\hat{\mathbf{e}}_{k+1} = \hat{\mathbf{x}}_{k+1} + r_{k+1}$, which implies $\mathbf{e}_{k+1} = \hat{\mathbf{e}}_{k+1} - \mathbf{t}_{k+1}$. Thus, equation (3.6) yields

$$\begin{aligned} \mathbf{u}_k^{mf} &= \arg \min_{\mathbf{u}_k} \{ (\hat{\mathbf{e}}_{k+1} - r_{k+1})^T \mathbf{W} (\hat{\mathbf{e}}_{k+1} - r_{k+1}) + \mathbf{u}_k^T \mathbf{Z} \mathbf{u}_k \} \\ &= \arg \min_{\mathbf{u}_k} \{ \hat{\mathbf{e}}_{k+1}^T \mathbf{W} \hat{\mathbf{e}}_{k+1} + r_{k+1}^T \mathbf{W} r_{k+1} + 2 \hat{\mathbf{e}}_{k+1}^T \mathbf{W} r_{k+1} + \mathbf{u}_k^T \mathbf{Z} \mathbf{u}_k \} \end{aligned} \quad (3.8)$$

Now, to establish a fair comparison between the model-free controller and the LQR controller, let $\mathbf{R} = \mathbf{Z}$ and

$\mathbf{Q} = \begin{pmatrix} \mathbf{W} & \mathbf{0} \\ \mathbf{0} & \mathbf{V} \end{pmatrix}$ in (3.1) where \mathbf{V} specifies the relative weight of cumulative errors. Thus, the LQR control law can be expressed as:

$$\begin{aligned} \{ \mathbf{u}^{LQR} \} &= \arg \min_{\{ \mathbf{u}_k \}} \sum_{k=1}^{N-1} \left\{ \begin{bmatrix} \hat{\mathbf{e}}_k^T & \hat{\mathbf{v}}_k^T \end{bmatrix} \begin{pmatrix} \mathbf{W} & \mathbf{0} \\ \mathbf{0} & \mathbf{V} \end{pmatrix} \begin{bmatrix} \hat{\mathbf{e}}_k \\ \hat{\mathbf{v}}_k \end{bmatrix} + \mathbf{u}_k^T \mathbf{Z} \mathbf{u}_k \right\} \\ &= \arg \min_{\{ \mathbf{u}_k \}} \sum_{k=1}^{N-1} \{ \hat{\mathbf{e}}_{k+1}^T \mathbf{W} \hat{\mathbf{e}}_{k+1} + \hat{\mathbf{v}}_{k+1}^T \mathbf{V} \hat{\mathbf{v}}_{k+1} + \mathbf{u}_k^T \mathbf{Z} \mathbf{u}_k \} \end{aligned} \quad (3.9)$$

Comparing (3.8) to (3.9), we observe that the model-free controller performs minimization *at each iteration*, whereas LQR performs a *single* minimization over all the iterations; each with respect to their individual loss functions. This particular phenomenon is rather a design issue. Whether the control engineer chooses to minimize error at each iteration or prefers minimizing the total sum of errors over the entire control horizon for a given particular control problem depends rather on the nature of the system being controlled and/or the specific goals of the control process.

Now, suppose that the system under consideration is linear and time-invariant with no impact of noise. In other words, $\phi_k(\mathbf{x}_k, \mathbf{u}_k, \mathbf{w}_k) = \mathbf{A} \mathbf{x}_k + \mathbf{B} \mathbf{u}_k$. Let $\bar{\mathbf{L}}_k := \mathbf{e}_{k+1}^T \mathbf{W} \mathbf{e}_{k+1} + \mathbf{u}_k^T \mathbf{Z} \mathbf{u}_k$ and $\mathbf{V} = \mathbf{0}$. Thus, $r_{k+1} = 0$ and $\hat{\mathbf{e}}_{k+1} = \mathbf{e}_{k+1}$ for all k , which yields

$$\begin{aligned} \mathbf{u}_k^{mf} &= \arg \min_{\mathbf{u}_k} \bar{\mathbf{L}}_k, \\ \{ \mathbf{u}^{LQR} \} &= \min_{\{ \mathbf{u}_k \}} \sum_{k=1}^{N-1} \bar{\mathbf{L}}_k. \end{aligned}$$

Thus, in the case of a linear system without any stochastic effects, the residual terms in (3.8) vanish and the loss function of the model-free controller becomes *equivalent* to that of the LQR controller with $\mathbf{V} = \mathbf{0}$. That is, both the model-free and LQR controllers would be minimizing the same loss function, where the model-free controller again would be executed at each iteration, while the LQR would be executed over all the iterations. Furthermore, assuming that all the target values are physically realizable and the NN structure in the model-free controller is capable of representing linear systems without any approximation errors, both controllers would yield the *same* control inputs, i.e., these two controllers would essentially be equivalent. Also notice that, in this particular case, the model-free controller would interestingly become a minimum variance controller. Note that a fundamental advantage of the model-free controller in general is that it requires only one-step-ahead target values, as opposed to LQR that requires a priori knowledge of the *entire* target sequence; the latter is a desirable feature in real-time control.

IV. ILLUSTRATION: WASTEWATER TREATMENT PROBLEM

A. Problem Description

The wastewater problem is described in [6] as follows: influent wastewater is first mixed (as determined by a controller) with a dilution substance to provide a mixture with a desired concentration of contaminants. This diluted

mixture is then sent to a second tank at a controlled flow rate. In the second tank the mixture goes through an anaerobic digestion process, where the organic material in the mixture is converted by bacteria into byproducts such as methane. Therefore, the system consists of two controls (the mix of wastewater/dilution substance and the input flow rate) and two states (an effluent de-polluted water and methane gas, which is useful as a fuel). Since this system relies on biological processes, the dynamics are nonlinear and usually time-varying. Also, the system is subject to constraints (e.g., the input and output concentrations, the methane gas flow rate, and the input flow rate all must be greater than zero), which presents an additional challenge in developing a controller for the system.

The unknown process equations are assumed to be

$$\begin{pmatrix} x_{k+1,1} \\ x_{k+2,2} \end{pmatrix} = \begin{pmatrix} 1 + \mu_k T & 0 \\ -0.3636T & 1 \end{pmatrix} \begin{pmatrix} x_{k,1} \\ x_{k,2} \end{pmatrix} + \begin{pmatrix} -Tx_{k,1} & 0 \\ -Tx_{k,2} & T \end{pmatrix} \begin{pmatrix} u_{k,1} \\ u_{k,1}u_{k,2} \end{pmatrix} + \begin{pmatrix} w_{k,1} \\ w_{k,2} \end{pmatrix} \quad (4.1a)$$

where the bacterial growth rate μ_k is given by

$$\mu_k = \frac{(0.425 + 0.025 \sin(2\pi k/96))x_{k,2}}{.4 + x_{k,2}} \quad (4.1b)$$

where

- x_1 is the methane gas flow rate,
- x_2 is the substrate output concentration,
- u_1 is wastewater/dilution substance mix rate,
- u_2 is the input flow rate, and
- T is the sampling interval, which is .5 seconds.

B. Model-Free Controller Implementation

We now replicate the problem environment and the model-free controller in [6]. The target sequence t_k is a periodic square wave with values (.97, .2) for the first 48 updates and (1, .1) for the second 48 updates¹. We assume independent noise terms $w_{k,1}$ and $w_{k,2} \sim N(0, \sigma^2 I)$ where $\sigma = .01$. The initial state is assumed to be $x_0 = (.5, 1.6375)$.

Note that the model-free controller has *no knowledge* of the above equations. The performance criteria used is the weighted root-mean-square (RMS) measurement:

$$\left[(x_{k+1} - t_{k+1})^T \mathbf{W} (x_{k+1} - t_{k+1}) + u_k^T \mathbf{Z} u_k \right]^{1/2} \quad (4.2)$$

with $\mathbf{W} = \text{diag}(.01, .99)$ and $\mathbf{Z} = \text{diag}(.001, .001)$ where $\text{diag}()$ denotes the square matrix whose diagonal entries are the given parameters and all off-diagonal entries are zero. Notice that (4.2) corresponds to the loss function (2.2) with the matrix $\mathbf{W}_k = \mathbf{W}$ and $\mathbf{Z}_k = \mathbf{Z}$. The values .01 and .99 reflect the relative emphasis of the controller on methane production and water purity, respectively. The control gains,

on the other hand, are weighted less compared to deviations from the target values. The NN used contains two hidden layers with 20 nodes in the first hidden layer and 10 nodes in the second. All the hidden nodes have the scaled logistic function as the activation function. The inputs to the NN are the current and most recent states, the most recent control, and the target vector for the next state, yielding a total of eight input nodes. The output of the NN is then the next control values. Thus, there are a total of 412 weights in the NN, which will be updated by SPSA at each iteration. These weights are initialized with random values in $[-.1, .1]$. As for the SPSA implementation, a two-sided SPSA with constant gains is used (since the system is time-varying) where $a = .2$ and $c = .1$. SPSA is implemented without any gradient approximation averaging or smoothing.

Figures 1a and 1b show the state values versus target values for each target-state pair when the model-free controller is used. The discrepancy between the tracking errors of the two state variables is not just a result of the weight emphasis we have put on the RMS loss function via matrix \mathbf{W} . In fact, this discrepancy is built into the control system proposed by Dochain and Bastin [7], whose research also showed that the system exhibits preferential tracking of one state variable over the other. From a physical perspective, this can be explained by the wastewater system having been designed to prioritize the de-polluted water output over that of methane gas through the parameters proposed in [6]. Indeed, changing the weights of the weight matrix \mathbf{W} offers slightly different results, but not by much, regardless of the weights used.

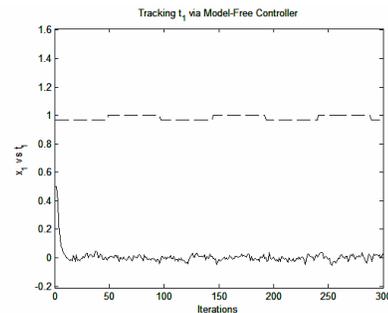


Fig. 1a: Model-free controller: tracking x_1 (solid line) versus t_1 (dashed line)

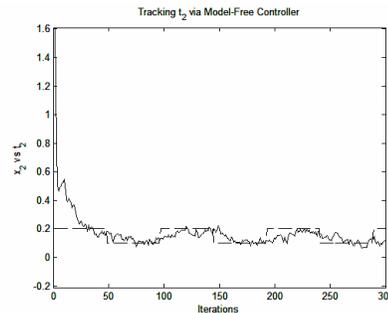


Fig. 1b: Model-free controller: tracking x_2 (solid line) versus t_2 (dashed line)

¹ Target values for x_2 in the first 48 updates are .13 in [2]. We use .2 to better illustrate the model-free controller's tracking capabilities.

C. Linear System Identification

We perform the system identification task via linear regression in two steps: collecting the data from which a model will be constructed and constructing an appropriate model from this data. For data collection, open-loop training with random inputs was performed where the bounds on the control inputs are $[.09, .4]$ for u_1 and $[1.5, 3.0]$ for u_2 (as in [7]); with the system initialized at $x_0 = (.5, 1.6375)$. We generated 300 random controls within those bounds and evaluated the noisy state values when the process is subject to these controls, obtaining 300 random input-output pairs. Having generated the data, we fitted a first-order linear time invariant auto-regressive (ARX) model, which is given by

$$\begin{pmatrix} x_{k+1,1} \\ x_{k+2,2} \end{pmatrix} = \mathbf{A} \begin{pmatrix} x_{k,1} \\ x_{k,2} \end{pmatrix} + \mathbf{B} \begin{pmatrix} u_{k,1} \\ u_{k,2} \end{pmatrix} \quad (4.3)$$

where the 2×2 matrices \mathbf{A} and \mathbf{B} are estimated using least squares regression. We chose the first-order model since it is simple and increasing the order did not significantly increase the model quality. For model evaluation, we computed the R^2 statistic associated with the regression, which revealed to be .98 for both x_1 and x_2 . Thus, a first-order linear model is quite good even though the underlying system is stochastic and nonlinear, which indicates validity of designing classical linear controllers for the wastewater problem. The least squares regression resulted in the following linear model:

$$\begin{pmatrix} x_{k+1,1} \\ x_{k+2,2} \end{pmatrix} = \begin{pmatrix} 1.0333 & .0907 \\ -.1786 & .8924 \end{pmatrix} \begin{pmatrix} x_{k,1} \\ x_{k,2} \end{pmatrix} + \begin{pmatrix} -.5204 & -.0007 \\ .7851 & .1172 \end{pmatrix} \begin{pmatrix} u_{k,1} \\ u_{k,2} \end{pmatrix} \quad (4.4)$$

D. LQR Implementation

In our LQR implementation, we chose $\mathbf{Q} = \begin{pmatrix} \mathbf{W} & \mathbf{0} \\ \mathbf{0} & \mathbf{V} \end{pmatrix}$ with $\mathbf{W} = \text{diag}(.01, .99)$ and $\mathbf{V} = (.01)\mathbf{W}$ (placing more emphasis on the control errors relative to the accumulated control errors) and $\mathbf{R} = \text{diag}(.001, .001)$. Notice that our choice of \mathbf{Q} and \mathbf{R} coincides with the model-free controller loss function for a fair comparison. Figures 2a & 2b illustrate the LQR controller performance.

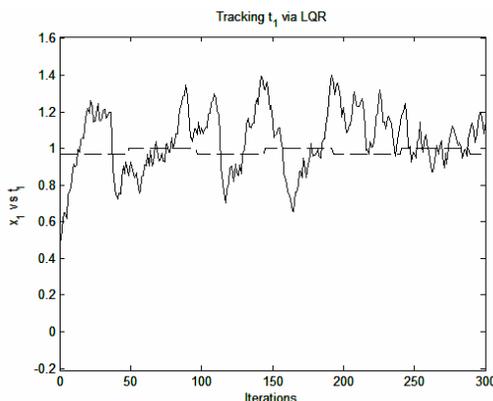


Fig. 2a: LQR controller: tracking x_1 (solid line) versus t_1 (dashed line)

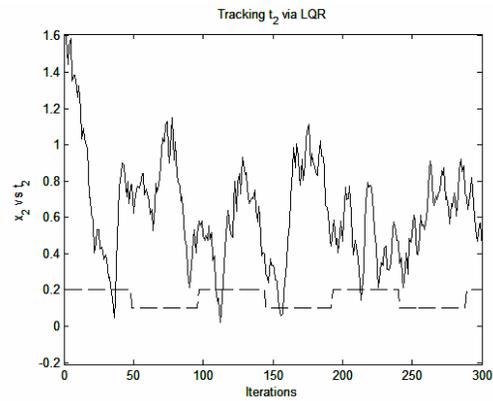


Fig. 2b: LQR controller: tracking x_2 (solid line) versus t_2 (dashed line)

As the above figures show, LQR outputs follow the target outputs to some degree, but not very closely; with t_1 being tracked somewhat better than t_2 . We tried fine-tuning the \mathbf{Q} and \mathbf{R} matrices, yet did not observe any significant performance improvements. In fact, the LQR controller exhibited preferential tracking for different target values that we tried, with t_1 being tracked better than t_2 and vice versa. We hypothesize that this occurs because the matrices that define the loss function J were not fine-tuned throughout the simulation process, as the input kept oscillating. That would have assumed a controller of the *adaptive* type, and is beyond our scope due to the fact that the model-free controller is not of the adaptive type either. Moreover, we possess no intuition as to how the \mathbf{Q} and \mathbf{R} matrices should be automatically updated as a function of changing target values.

Analysis of the LQR controller output reveals that the state values are in the range of the target values, yet x_2 is a lot more amplified than x_1 . However, the LQR controller still behaves in a far worse fashion than the model-free controller. Since we have formulated both controllers to minimize similar loss functions, the difference between the behaviors of the two controllers can be attributed to the way each controller handles iteration error. The model-free controller updates itself after each iteration, thereby keeping the error between input and output to a minimum. On the contrary, the LQR algorithm sums the error over the whole simulation process, only attempting minimization at the very end. It is precisely this buildup in error that prevents the LQR controller from tracking as well as the model free SPSA controller. This may be particular to the nature of this MIMO system and how disturbances in one state variable affect the other state variable through relationships to be found in the system's state space. Therefore, in this application, the inability of the LQR controller to compensate for the error (between the actual output and target output) quickly enough actually penalizes it and forces the tracking to deteriorate.

V. SUMMARY & CONCLUSIONS

In this paper, we discuss model-free and classical controllers for the control of nonlinear stochastic systems and briefly describe their basic characteristics and limitations. We then present a limited formal comparison in the case of LQR controllers. Specifically, we show that both controllers are governed by the same mathematical models; the difference being the way each controller handles error propagation. Furthermore, given a previously defined wastewater treatment problem by Dochain and Bastin [7], and a solution to the MIMO control of this system implemented by Spall and Cristion [6] through SPSA, we attempt to solve this problem through a linear quadratic regulator (LQR). The implementation of the LQR controller for this problem allowed us to study the interesting coupling between the two states of the system and observe some interesting comparisons between LQR and model-free controllers. These comparisons are more insightful because both controllers incorporate a minimization function, which tailors their respective outputs accordingly. However, LQR has one disadvantage over the model-free controller in that it is model based and thus constrained by the values of the state equations by which the model is described. Furthermore, it was observed that the summation of error on behalf of the LQR controller actually makes it perform worse than the model-free controller, which looks at error at every iteration of the system. This gives the model-free controller the flexibility to adapt to changes in a monitored system, its only limit being the definition of its loss function. A comparison of the two algorithms shows that both choose to track this MIMO system preferentially; that is, tracking of a certain variable is prioritized over the tracking of the other. However, each algorithm “chooses” to do so differently. The model-free approach matches the gains of the desired outputs but offsets one of them at the cost of following the other. The LQR regulator matches the overall value of the outputs well, but “chooses” to have smaller steady state error on one output, at the cost of the other, again an effect of error summation rather than adaptation per iteration.

Another distinct advantage of the model-free controller is that it can attempt to control systems whose internal processes cannot be observed because of real world constraints. The model-free controller will assume a solution as long as it is mathematically possible. While this is advantageous to the designer, it is a tool that must be used carefully. In control systems design, the state equations are designed based on measured parameters of sensors and the physical properties of the components. Thus a user of the model-free controller will have to choose the cost function for this algorithm very wisely, to make sure that, if used in a design tool, certain physical properties are met, such as controllability and stability of a physical system.

VI. DIRECTIONS FOR FUTURE RESEARCH

The comparison of the model-free controller to LQR can be extended to formally account for stochastic effects and/or incorporate linearization error for certain classes of nonlinear dynamical systems. The model-free controller can further be compared to other methods of classical control, such as linear quadratic Gaussian (LQG) controllers or control via pole-placement. In addition, the model-free controller can be compared to neural network-based controllers as in [17], which would provide significant information on the relative value of utilizing truly model-free controllers versus first constructing a neural network representation of the system being controlled. These comparisons should be made generically to the extent possible, including effects of process and/or measurement noise.

REFERENCES

- [1] Friesz, T. L., Luque, J., Tobin, R. L., Wie B-W., “Dynamic network traffic assignment considered as a continuous time optimal control problem”, *Operations Research*, 37(6), pp. 893-901.
- [2] Taylor, H. R., Funda, J., Eldridge, B., Gomory, S., Gruben, K., LaRose, D., Talamini, M., Kavoussi, L., Anderson, J., “A telerobotic assistant for laparoscopic surgery”, *IEEE Engineering in Medicine and Biology Magazine*, 14(3), pp. 279-288.
- [3] Phillips, D. J., *Quantitative Analysis in Financial Markets*, NJ: Hackensack, World Scientific, 1999.
- [4] Khalil, H. K., *Nonlinear Systems*, 3rd Ed., NJ: Englewood Cliffs, Prentice Hall, 2002.
- [5] Spall, J. C. and Cristion, J. A., “Model-free control of nonlinear systems with discrete time measurements,” *IEEE Transactions on Automatic Control*, vol. 43, pp. 1198–1210, 1998.
- [6] Spall, J. C. and Cristion, J. A., “A neural network controller for systems with unmodeled dynamics with application to wastewater treatment,” *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 27, pp. 369–375, 1997.
- [7] Dochain, D. and Bastin, G., “Adaptive identification and control algorithms for nonlinear bacterial growth systems,” *Automatica*, vol. 20, pp. 621–634, 1984.
- [8] Sadegh, P. and Spall, J. C., “Optimal sensor configuration for complex systems”, *Proceedings of the American Control Conference*, 1998.
- [9] Maeda, Y. and De Figueiredo, R. J. P., “Learning rules for neuro-controller via simultaneous perturbation”, *IEEE Transactions on Neural Networks*, vol. 8, pp. 1119–1130, 1997.
- [10] Song, J., Xu, Y., Yam, Y., and Nechyba, M. C., “Optimization of human control strategy with simultaneously perturbed stochastic approximation”, *Proceedings of the IEEE Conference on Intelligent Robots and Systems*, 1998.
- [11] Vande Wouwer, A., Renotte, C., Bogaerts, P., and Remy, M., “Application of SPSA techniques in nonlinear system identification”, *European Control Conference*, 2001.
- [12] Ji, X.D. and FAMILONI, B.D., “A diagonal recurrent neural network-based hybrid direct adaptive SPSA control system,” *IEEE Transactions on Automatic Control*, vol. 44, pp. 1469-1473, 1999.
- [13] Ogata, K., *Discrete-Time Control Systems*, 2nd Ed., NJ: Englewood Cliffs, Prentice Hall, 1987.
- [14] Astrom, J.K., *Computer Controlled Systems*, NJ: Englewood Cliffs, Prentice Hall, 1990.
- [15] Franklin, G.F., Powell, J.D., and Workman, M.L., *Digital Control of Dynamic Systems*, 3rd Ed., NJ: Englewood Cliffs, Prentice Hall, 1997.
- [16] Ljung, L., *System Identification: Theory For the User*, 2nd Ed., NJ: Englewood Cliffs, Prentice Hall, 1998.
- [17] Narendra, K. S. and Parthasarathy, K., “Identification and control of dynamical systems using neural networks”, *IEEE Transactions on Neural Networks*, vol. 1, pp.4-26, 1990.