A Matrix-T Approach to the Sequential Design of Optimization Experiments

Enrique del Castillo*

Eduardo Santiago

Dept. of Industrial Engineering, Pennsylvania State University, University Park, PA 16802

May 5, 2009

*Corresponding author. E-mail exd13@psu.edu

Abstract

A new approach to the sequential design of experiments for the rapid optimization of multiple response, multiple controllable factor processes, is presented. The approach is Bayesian, and approximates the cost to go function of the underlying Dynamic Programming problem. The approximation is based on a matrix T posterior predictive density for the predicted responses which can be cross-correlated and/or correlated over time. The case of an unknown variance is addressed; the assumed models are linear in the parameters but can be nonlinear in the factors. It is shown that the proposed approach has dual control features, initially probing the process to reduce the parameter uncertainties and eventually converging to the desired optimal solution. The convergence of the proposed method is numerically studied and convergence conditions discussed. A computer implementation of the algorithm is provided.

Keywords: Response Surface Methods, Dual Control, Cost to go, Dynamic Programming.

1 Introduction

In a sequential Design of Experiments (DOE), the experimental conditions, given by the levels of the controllable factors in each run $\boldsymbol{u}' = (u_1, u_2, ..., u_q)$, are selected based on the levels used in previous runs and on the previous observed response values. Formal algorithmic approaches to the sequential design of experiments are of practical value when the purpose of the experiment is to optimize a process or system as rapidly as possible, e.g., when the cost of running an experiment is extremely high and the cost of poor response values unacceptable. These conditions are frequent in capital-intensive industries (e.g., semiconductor, large chemical plants). This paper presents a new approach to sequential DOE for the rapid optimization of multiple response processes where the input-output relation is given by a multivariate statistical model linear in the parameters which are also unknown.

The sequential nature of experimentation is frequently acknowledged in the Engineering Statistics literature (e.g., Box, Hunter and Hunter, 2005). Early approaches to sequential DOE include the Evolutionary Operation (EVOP) technique for on-line optimization of a single-response process (Box and Draper, 1957) and the work by Kiefer and Wolfowitz (1952), who adopted Robbins and Monro's landmark stochastic approximation algorithm to the problem of finding the stationary point of a single factor regression function (see Robbins and Monro 1951), an approach soon extended by Blum (1954) to the multiple factor case. This line of work has received considerable impetus in recent years originating from Spall's stochastic perturbation methods (see e.g. Spall 1992 and 2005) which require only two experiments per iteration in the multiple factor case. However, the total number of

iterations needed for convergence is too large to be practical in real industrial experiments. Furthermore, these approaches do not fit a model.

Model-based sequential DOE for the optimization of a process has been discussed in the control literature. Feldbaum (1960) introduced the concept of *dual control*: to control an unknown system a sequential algorithm must initially "probe" the system by varying the controllable factors $\boldsymbol{u}$ to learn about (or estimate) some postulated process model. Once enough information is gathered a sequential strategy should then lead $\boldsymbol{u}$ to a region were the control objective for the response(s) of interest is achieved. The problem then has two conflicting objectives, estimation and control. The same underlying dual control strategy is necessary for *optimization* objectives, leading to a large literature, too vast to review here in detail, on Optimizing Control, Self-tuning Extremum Control, and related methods (see e.g. Zarrop 1991). These methods fall under the general category of Adaptive Control, where the algorithms act as the *controller* of a process (which determines the next run in the sequence), a terminology not commonly found in DOE. Adaptive Dual Control problems are formulated as Dynamic Programming (DP) problems for which exact solutions are impossible to find, even for very small problems, due to curse-of-dimensionality problems that plague this approach (e.g. see Åström and Wittenmark, 1989).

Approximate but faster solutions that try to mimic the "dual features" (estimation and control) of the optimal control policy have been proposed. A popular approach is that by Milito et al. (1982), (Milito (1982), (see also Wittenmark, 1975) who suggested a two-component objective function that considers the dual objectives explicitly:

$$\min_{\boldsymbol{u}_t} \ (\widehat{y}_{t+1} - \tau)^2 - \omega \frac{|\boldsymbol{P}_{t+1}|}{|\boldsymbol{P}_{t+2}|} \tag{1}$$

where $\boldsymbol{P}_t$ is the scaled covariance matrix of the parameter estimates used to compute the predicted response $\widehat{y}$ and τ is the response target (one response assumed). If $\omega = 1$, then the objective favors good parameter estimates [the second term is similar to the conditional D-optimality criterion (see, e.g., Del Castillo, 2008)]; if $\omega = 0$ the problem favors optimization neglecting the uncertainty of the parameter estimates (certainty equivalence controller). Pronzato (2000) shows how using a constant ω in (1) does not satisfy the sufficient convergence conditions of adaptive control algorithms, first established by Lai and Wei (1982).

Ben-Gal and Caraminis (2002) recently propose a DP-based strategy for a single response system that explicitly considers the monetary cost of each experiment. Their approach aims to find experimental designs to best estimate (gain information about) a process model. Notable features of their approach are the consideration of a model updating step via lack of fit tests and the development of rules that indicate when to stop experimenting.

A different attack on the Dual Control problem, adopted in this paper to generate a sequential DOE, is to approximate the "cost to go" equation used by the exact solution in

the DP formulation. First explored in Tse and Bar-Shalom (1973), this strategy has received definite recent contributions by Pronzato and coworkers, who studied the case of a *single* response process and a *known* process variance (see Kulcsar et al., 1996).

This paper addresses the unknown variance, multiple response, multiple factor case, when the input-output relation may be nonlinear in $\boldsymbol{u}$. The approach is Bayesian, and the approximated cost to go function requires the use of a matrix T predictive density. This density allows to model correlation between the responses (a common concern in the DOE literature) and also allows to model correlation over time. Pronzato et al. (1995) have shown how even if a process obeys a static model, the dual control solution constitutes a dynamic system given the interplay between parameter estimation and control (i.e., correlation between parameter estimates obtained close results in response values correlated over time, etc.).

The rest of this paper is organized as follows. Sections 2 and 3 discuss the assumed statistical model and cost function, respectively. Section 4 presents the approximated cost to go function. Section 5 explains the prior-posterior computations (matrix T) used by the proposed solution. Section 6 the numerical behavior of the proposed sequential DOE approach is studied via simulation. Comparisons are made with an off-line, known-parameters solution (which provides a best performance benchmark) and with the case where the dual control algorithm is given initially perfect information about the process parameters. Finally, section 7 explores the convergence properties of the proposed method. The paper ends with conclusions and directions for further research.

2 Model assumptions

Consider the following multivariate linear statistical model:

$$\boldsymbol{y} = \boldsymbol{\Theta}\boldsymbol{x} + \boldsymbol{\varepsilon}, \quad \boldsymbol{\varepsilon} \sim N(\boldsymbol{0}, \boldsymbol{V}) \tag{2}$$

where $\boldsymbol{y}$ is a $p \times 1$ vector of responses, $\boldsymbol{x}$ is an $r \times 1$ vector of regressors and $\boldsymbol{V}$ is an unknown $p \times p$ covariance matrix. Let $\boldsymbol{u}$ be a $q \times 1$ vector of controllable factors so that $\boldsymbol{x} = \boldsymbol{f}(\boldsymbol{u})$. The nature of $\boldsymbol{f}(\boldsymbol{u})$ can be nonlinear in $\boldsymbol{u}$ as long as $\boldsymbol{y}$ is linear in the parameters. We focus in this paper on the case when $\boldsymbol{f}(\boldsymbol{u})$ corresponds to a second order polynomial model:

$$\boldsymbol{x} = \boldsymbol{f}(\boldsymbol{u}) = [1, u_1, u_2, ..., u_q, u_1u_2, ..., u_{q-1}u_q, u_1^2, ..., u_q^2]'. \tag{3}$$

thus $r = \dim(\boldsymbol{x}) = (q+2)(q+1)/2$ in this case. The $p \times r$ matrix $\boldsymbol{\Theta}$ contains unknown model parameters that correspond to the model form specified by $\boldsymbol{x}$.

In the sequential design problems discussed herein it will be necessary to predict the p responses for the next m periods based on k observations. Let $\tilde{\boldsymbol{Y}} = [\tilde{\boldsymbol{y}}_{k+1}, \ldots, \tilde{\boldsymbol{y}}_{k+m}]$ (from here on, a tilde over a variable indicates a future, non yet observed variable). As discussed by

a number of authors (see Geisser, 1993, Kibria, 2006, and Minka, 2001), the predictive density of $\tilde{\boldsymbol{Y}}$, either for non-informative or conjugate/informative priors is a matrix T distribution.

A $p \times m$ random matrix $\boldsymbol{T}$ is said to follow a non-central matrix T distribution (denoted $\boldsymbol{T} \sim \boldsymbol{T}_{p,m}(\nu, \boldsymbol{\mu}, \boldsymbol{\Psi}, \boldsymbol{\Xi})$) if its density satisfies:

$$p(\boldsymbol{T}) \propto |\boldsymbol{\Psi}^{-1} + (\boldsymbol{T} - \boldsymbol{\mu})\boldsymbol{\Xi}^{-1}(\boldsymbol{T} - \boldsymbol{\mu})'|^{-\frac{\nu+p+m-1}{2}}$$

where $\boldsymbol{\Psi} > 0$ is a $p \times p$ matrix, $\boldsymbol{\Xi} > 0$ is $m \times m$, and $\nu > p + m - 1$ is a shape parameter (the degrees of freedom). The mean and variance of $\boldsymbol{T}$ are $E[\boldsymbol{T}] = \boldsymbol{\mu}$ and $\text{Var}(\boldsymbol{T}) = \boldsymbol{\Xi} \otimes \boldsymbol{\Psi}/(\nu - 2)$, which exists if $\nu > 2$. Associated with a random matrix $\boldsymbol{T}$ are *two* covariance matrices, $\boldsymbol{\Psi}$ and $\boldsymbol{\Xi}$. The correlation between the rows of $\boldsymbol{T}$ is modeled by $\boldsymbol{\Psi}$. The correlation between the columns of $\boldsymbol{T}$ is modeled by $\boldsymbol{\Xi}$. If either $\boldsymbol{\Psi}$ or $\boldsymbol{\Xi}$ is diagonal it implies the rows or columns are uncorrelated. Thus, the predictions may be correlated across the m future time periods or correlation can exist contemporaneously between the p predicted responses.

3 Assumed cost function

Let the cost of running the process at period $k+1$ with controllable factors $\boldsymbol{u}_k$, set at period k, be the quadratic function:

$$c_k(\boldsymbol{u}_k) = (\tilde{\boldsymbol{y}}_{k+1} - \boldsymbol{\tau}_{k+1})'\boldsymbol{\Gamma}_{k+1}(\tilde{\boldsymbol{y}}_{k+1} - \boldsymbol{\tau}_{k+1}) + \boldsymbol{w}_k'\boldsymbol{R}_k\boldsymbol{w}_k \tag{4}$$

where $\tilde{\boldsymbol{y}}_{k+1}$ is a $p \times 1$ random vector of one step ahead future responses with corresponding vector of targets $\boldsymbol{\tau}_{k+1}$, $\boldsymbol{\Gamma}_{k+1}$ is a $p \times p$ matrix of off-target costs (usually a diagonal matrix), $\boldsymbol{w}_k = \boldsymbol{u}_k - \boldsymbol{u}_{k-1}$ is the adjustment made to the factors and $\boldsymbol{R}_k$ is a $q \times q$ adjustment cost matrix (also diagonal).

The objective to minimize is the sum of all expected costs involved over a horizon of N observations or periods, that is,

$$J_N(\boldsymbol{U}^{(N)}) = E\left[\sum_{i=1}^{N} c_i(\boldsymbol{u}_i)|D^{(0)}\right] \tag{5}$$

where $\boldsymbol{U}^{(N)} = \{\boldsymbol{u}_1, \boldsymbol{u}_2, ..., \boldsymbol{u}_N\}$. Note that $c_N(\boldsymbol{u}_N) = 0$, as there is no adjustment in period N. $D^{(i)}$ denotes the information available at the end of period i, so we have that

$$D^{(j)} = \left\{\boldsymbol{U}^{(j-1)}, \boldsymbol{Y}^{(j)}, D^{(0)}\right\}$$

where $\boldsymbol{Y}^{(j)} = \{\boldsymbol{y}_1, \boldsymbol{y}_2, ..., \boldsymbol{y}_j\}$ and $\boldsymbol{U}^{(0)} = \emptyset$.

The Dynamic Programming equation that solves problem (5) is

$$\begin{aligned}\min_{\boldsymbol{u}_1} E_{\tilde{\boldsymbol{y}}_2}\{c_1(\boldsymbol{u}_1) + \cdots + \min_{\boldsymbol{u}_k} E_{\tilde{\boldsymbol{y}}_{k+1}}[c_k(\boldsymbol{u}_k) + \min_{\boldsymbol{u}_{k+1}} E_{\tilde{\boldsymbol{y}}_{k+2}}[c_{k+1}(\boldsymbol{u}_{k+1}) + \cdots \\ \cdots + \min_{\boldsymbol{u}_{N-1}} E_{\tilde{\boldsymbol{y}}_N}[c_{N-1}(\boldsymbol{u}_{N-1})|D^{(N-1)}] \cdots |D^{(k+1)}]|D^{(k)}] \cdots |D^{(1)}\}\end{aligned} \tag{6}$$

The optimal cost to go at step k is given by the Bellman equation:

$$J^*_{k,N}(D^{(k)}) = \min_{\boldsymbol{u}_k} E_{\tilde{\boldsymbol{y}}_{k+1}} \left[c_k(\boldsymbol{u}_k) + J^*_{k+1,N}(D^{(k+1)}) | D^{(k)} \right], \quad k = 1, 2, ..., N-1$$

with boundary condition $J^*_{N,N}(D^{(N)}) = 0$. No simple recursive solution exists for this equation when $\boldsymbol{\Theta}$ and $\boldsymbol{V}$ are unknown. When they are known and $\boldsymbol{f}(\boldsymbol{u}_i)$ is linear in $\boldsymbol{u}$, the Bellman equation has a well known solution, the Linear Quadratic Gaussian (LQG) solution described in Åström (1970). In this paper, in contrast, $\boldsymbol{f}(\boldsymbol{u}_i)$ is assumed to be quadratic and the parameters $\boldsymbol{\Theta}$ and $\boldsymbol{V}$ are assumed to be unknown, and hence an approximation to the cost to go is necessary.

4 An approximation to the cost to go

Following the notation in the last section, the cost to go at the end of period k can be expressed as follows:

$$J_{k,N}(\boldsymbol{u}_k) = E_{\tilde{\boldsymbol{y}}_{k+1}}[c_k(\boldsymbol{u}_k) + J^*_{k+1,N}(D^{(k+1)}) | D^{(k)}] \tag{7}$$

which needs to be minimized with respect to $\boldsymbol{u}_k, \boldsymbol{u}_{k+1}, \ldots, \boldsymbol{u}_{N-1}$. To evaluate the right hand side, one needs:

- The predictive density $\tilde{\boldsymbol{y}}_{k+1} | (D^{(k)}, \boldsymbol{u}_k)$ for the first term, and
- The predictive densities $\tilde{\boldsymbol{y}}_{k+2} | (D^{(k+1)}, \boldsymbol{u}_{k+1}), \ldots, \tilde{\boldsymbol{y}}_N | (D^{(N-1)}, \boldsymbol{u}_{N-1})$ for the second term.

The difficulty of the problem lies in that the expected value in equation (7) depends on future observations that in turn depend on yet unobserved factor settings $\boldsymbol{u}_j, j > k$, i.e., $D^{(k)}$ contains $\boldsymbol{u}_k$, $D^{(k+1)}$ contains $\boldsymbol{u}_{k+1}$, etc.. Thus, finding the optimal cost to go $J^*_{k,N}(\boldsymbol{u}_k)$ requires optimization over nested multivariate expected values.

Following an idea in Kulcsar et al. (1996), who considered the case $p = q = 1$ and *known* variance, we approximate the predictive densities shown above using the information available at period k. Let $\widehat{\boldsymbol{\Theta}}_j$ denote the mean of $p(\boldsymbol{\Theta}|D^{(j)})$, the posterior of $\boldsymbol{\Theta}$ at period j. The predictive densities are then approximated as follows:

$$\begin{aligned} &\left\{ \tilde{\boldsymbol{y}}_{k+1} | (\widehat{\boldsymbol{\Theta}}_k, \boldsymbol{u}_k), \tilde{\boldsymbol{y}}_{k+2} | (\widehat{\boldsymbol{\Theta}}_{k+1}, \boldsymbol{u}_{k+1}), \ldots, \tilde{\boldsymbol{y}}_N | (\widehat{\boldsymbol{\Theta}}_{N-1}, \boldsymbol{u}_{N-1}) \right\} \\ \approx &\left\{ \tilde{\boldsymbol{y}}_{k+1} | (\widehat{\boldsymbol{\Theta}}_{k-1}, \boldsymbol{u}_k), \tilde{\boldsymbol{y}}_{k+2} | (\widehat{\boldsymbol{\Theta}}_{k-1}, \boldsymbol{u}_{k+1}), \ldots, \tilde{\boldsymbol{y}}_N | (\widehat{\boldsymbol{\Theta}}_{k-1}, \boldsymbol{u}_{N-1}) \right\} \end{aligned} \tag{8}$$

In Kulcsar et al. (1996), the authors noted that if the error variance is known (as assumed by them), then the posterior covariance matrix $\boldsymbol{P}_j$ of $p(\boldsymbol{\Theta}|D^{(j)})$ can be propagated forward in time via a Kalman filter and used in the approximate solution since it does not depend on

the data (past or future). This, unfortunately, is not the case in our present problem, since the covariance $\boldsymbol{V}$ is assumed unknown. Suppose for a moment, however, that the covariance $\boldsymbol{V}$ is known. The approximate cost to go associated with future process adjustments $\tilde{\boldsymbol{W}} = [\tilde{\boldsymbol{w}}_k, \ldots, \tilde{\boldsymbol{w}}_{N-1}]$ is equal to

$$\begin{aligned}\widehat{J}_{k,N}(\tilde{\boldsymbol{W}}) &= E\left[\sum_{i=k}^{N-1}(\tilde{\boldsymbol{y}}_{i+1}-\boldsymbol{\tau}_{i+1})'\boldsymbol{\Gamma}_{i+1}(\tilde{\boldsymbol{y}}_{i+1}-\boldsymbol{\tau}_{i+1})+\tilde{\boldsymbol{w}}_i'\boldsymbol{R}_i\tilde{\boldsymbol{w}}_i|D^{(i)}\right] \\ &= \sum_{i=k}^{N-1}(\widehat{\boldsymbol{\Theta}}_{k-1}\tilde{\boldsymbol{x}}_i-\boldsymbol{\tau}_{i+1})'\boldsymbol{\Gamma}_{i+1}(\widehat{\boldsymbol{\Theta}}_{k-1}\tilde{\boldsymbol{x}}_i-\boldsymbol{\tau}_{i+1})+tr(\boldsymbol{\Gamma}_{i+1}\mathrm{Var}(\tilde{\boldsymbol{y}}_{i+1}))+\tilde{\boldsymbol{w}}_i'\boldsymbol{R}_i\tilde{\boldsymbol{w}}_i\end{aligned}$$

where $\mathrm{Var}(\tilde{\boldsymbol{y}}_{i+1}) = \boldsymbol{V}\left\{1+\tilde{\boldsymbol{x}}_i'\boldsymbol{P}_{i-1}^{-1}\tilde{\boldsymbol{x}}_i\right\}$. If $\boldsymbol{V}$ is known, $\boldsymbol{P}_j$ does not depend on the future data.

Suppose now that $\boldsymbol{V}$ is unknown. Instead of considering the distributions in (8), consider the *joint* distribution of $\tilde{\boldsymbol{y}}_{k+1},\ \tilde{\boldsymbol{y}}_{k+2}, \ldots,\ \tilde{\boldsymbol{y}}_N$. This can be obtained from the $p \times m$ matrix (where $m = N-k$ is the number of periods to go):

$$\tilde{\boldsymbol{Y}} = [\tilde{\boldsymbol{y}}_{k+1}, \ldots, \tilde{\boldsymbol{y}}_N].$$

For an unknown variance $\boldsymbol{V}$ and normally distributed errors, the posterior predictive distribution of $\tilde{\boldsymbol{Y}}$ is a matric-variate T, see Kibria (2006) and Geisser (1993). With this random matrix, the approximated cost to go is

$$\widehat{J}_{k,N}(\tilde{\boldsymbol{W}}) = trE[(\tilde{\boldsymbol{Y}}-\boldsymbol{\tau})'\boldsymbol{\Gamma}(\tilde{\boldsymbol{Y}}-\boldsymbol{\tau})+\tilde{\boldsymbol{W}}'\boldsymbol{R}\tilde{\boldsymbol{W}}] \tag{9}$$

where $\boldsymbol{\tau}$ is a $p \times m$ matrix of response targets. Thus the approximate cost to go involves the expectation of a *matrix* quadratic form of a matrix T distribution.

Gupta and Nagar (2000) (p. 60 and p. 136), provide several results on matrix distributions useful to compute the approximate cost $\widehat{J}_{k,N}$. In particular, they show that if $\tilde{\boldsymbol{Y}} \sim \mathbf{T}_{p,m}(v, \boldsymbol{\mu}, \boldsymbol{P}, \boldsymbol{Q})$, then:

$$E[(\tilde{\boldsymbol{Y}}-\boldsymbol{\tau})'\boldsymbol{\Gamma}(\tilde{\boldsymbol{Y}}-\boldsymbol{\tau})] = \frac{1}{\nu-2}tr(\boldsymbol{\Gamma}'\boldsymbol{P})\boldsymbol{Q}+(\boldsymbol{\mu}-\boldsymbol{\tau})'\boldsymbol{\Gamma}(\boldsymbol{\mu}-\boldsymbol{\tau}) \tag{10}$$

where ν are the degrees of freedom of the matrix T distribution and $\boldsymbol{\mu} \doteq E[\tilde{\boldsymbol{Y}}|D^{(k)}] = \widehat{\boldsymbol{\Theta}}_{k-1}\tilde{\boldsymbol{X}}$. Combining (9) and (10), we obtain a closed-form expression for the approximate cost to go from period k to period N. The proposed approximate solution to problem (5) is then:

```
for k = 1 to N
    m = N − k
    min Ĵ_{k,N}(W̃) to get u_k, u_{k+1}, ..., u_{N−1}
    implement controllable factors u_k
    observe y_{k+1} and update D^{(k+1)}
end
```

The Bayesian update of the matrix T predictive density necessary to carry out the minimization of $\widehat{J}_{k,N}$ at each k is discussed next.

5 Assumed Priors and Bayesian update of the Posterior Predictive density

It is well-known (see Press, 1982) that the joint conjugate prior for $\boldsymbol{\Theta}$ and $\boldsymbol{V}$ under normal distributed errors is a Matrix Normal-Inverse Wishart, that is,

$$p(\boldsymbol{\Theta}, \boldsymbol{V}|\boldsymbol{X}^{(t)}) \sim N(\boldsymbol{\Theta}; \boldsymbol{\Theta}_0, \boldsymbol{V}, \boldsymbol{K}) \ \cdot \ W^{-1}(\boldsymbol{V}; \boldsymbol{S}_0, N_0)$$

where $\boldsymbol{\Theta}_0$, $\boldsymbol{K}, \boldsymbol{S}_0$ and N_0 are the prior parameters. Assume in particular that $\boldsymbol{\Theta}_0 = \boldsymbol{0}$, $\boldsymbol{K} = \alpha \boldsymbol{I}_r$, and $\boldsymbol{S}_0 = N_0 \boldsymbol{I}_p$. As $\alpha \to 0$ and $N_0 \to 0$ this prior becomes Jeffrey's non-informative prior, (see Minka, 2001). Under this prior, the posterior predictive density of a set of m future observation vectors is a matrix T distribution (Minka 2001, Kibria 2006):

$$\tilde{\boldsymbol{Y}} \sim \mathbf{T}_{p,m}(v, \boldsymbol{\mu}, \boldsymbol{P}, \boldsymbol{Q}),$$

with $\nu = k + N_0 - p + 1$ degrees of freedom. Here, k is the size of the data set, and N_0 can be understood as the number of periods worth of data the user thinks his/her prior is worth. The $p \times p$ matrix $\boldsymbol{P}$ is the variance-covariance matrix for the *rows* of $\tilde{\boldsymbol{Y}}$ and the $m \times m$ matrix $\boldsymbol{Q}$ is the covariance matrix for the *columns* of $\tilde{\boldsymbol{Y}}$. Let $\boldsymbol{X}^{(t)} = [\boldsymbol{x}_1, ..., \boldsymbol{x}_t]$ be a $r \times t$ matrix of past regressors. The covariance matrices can be updated as follows:

$$\boldsymbol{Q} = (\boldsymbol{I}_m - \tilde{\boldsymbol{X}}'(\boldsymbol{S}_{xx} + \tilde{\boldsymbol{X}}\tilde{\boldsymbol{X}}')^{-1}\tilde{\boldsymbol{X}})^{-1}$$

where $\boldsymbol{S}_{xx} = \boldsymbol{X}^{(t)}\boldsymbol{X}^{(t)\prime} + \boldsymbol{K}$. Note how as long as $\alpha > 0$ this prior allows to make inferences for all k, since $\boldsymbol{S}_{xx}$ will always be invertible. We also have that

$$\boldsymbol{P} = \boldsymbol{S}_{y|x} + \boldsymbol{S}_0$$

where $\boldsymbol{S}_{y|x} = \boldsymbol{S}_{yy} - \boldsymbol{S}_{yx}\boldsymbol{S}_{xx}^{-1}\boldsymbol{S}_{yx}'$, $\boldsymbol{S}_{yy} = \boldsymbol{Y}^{(t)}\boldsymbol{Y}^{(t)\prime} + \boldsymbol{\Theta}_0\boldsymbol{K}\boldsymbol{\Theta}_0'$, and $\boldsymbol{S}_{yx} = \boldsymbol{Y}^{(t)}\boldsymbol{X}^{(t)\prime} + \boldsymbol{\Theta}_0\boldsymbol{K}$. With the previous notation, let

$$\widehat{\boldsymbol{\Theta}}^{(t)} = \boldsymbol{S}_{yx}\boldsymbol{S}_{xx}^{-1}$$

so that $\boldsymbol{\mu} = E[\tilde{\boldsymbol{Y}}] = \widehat{\boldsymbol{\Theta}}^{(t)}\tilde{\boldsymbol{X}}$ represents the $p \times m$ mean matrix of $\tilde{\boldsymbol{Y}}$. We also have that $\text{Var}(\tilde{\boldsymbol{Y}}) = \boldsymbol{Q} \otimes \boldsymbol{P}/(\nu - 2)$, which exists only if $\nu > 2$, or equivalently, if $k > p + 1 - N_0$. Thus we have that

$$\tilde{\boldsymbol{Y}} - \boldsymbol{\tau} \sim \mathbf{T}_{p,m}(v, \boldsymbol{\mu} - \boldsymbol{\tau}, \boldsymbol{P}, \boldsymbol{Q}).$$

Thus, the approximate cost to go $\widehat{J}_{k,N}$ (equation 9) can be computed at each stage k using (10) and the results in this section.

6 Examples

Example 1. No adjustment cost, short horizon. Khuri (1996) reports the following models fitted from a $p = 2$, $q = 3$ experiment ran in a semiconductor manufacturing process:

$$\begin{aligned} y_1 &= [2756.5, 547.6, 616.3, \text{-}126.7, \text{-}52.9, \text{-}156.9, \text{-}550.3, \text{-}1109.5, \text{-}286.1, 989.1] \cdot \boldsymbol{x} + \varepsilon_1 \\ y_2 &= [746.3, 62.3, 128.6, \text{-}152.1, \text{-}28.9, \text{-}122.1, \text{-}140.6, \text{-}289.7, \text{-}32.1, 237.7] \cdot \boldsymbol{x} + \varepsilon_2 \end{aligned}$$

where $\boldsymbol{x}$ is defined as in equation (3) and $\boldsymbol{V}$ is diagonal with elements 60^2 and 30^2. Here, y_1 is the removal rate of metal in a chemical-mechanical polishing process, y_2 represents the within wafer standard deviation, and the controllable factors are Tablespeed (u_1), Downforce (u_2) and Slurry Concentration (u_3). The controllable factors have been coded into the (-1, 1) range following common practice in designed experiments. The desired targets for these responses are $\boldsymbol{\tau} = (3200, 500)'$. The second target is not achievable but was used in an effort to make the removal rate as large as possible. In this example we assume these models to be the true description of the process for simulation purposes, and assume no adjustment costs ($\boldsymbol{R} = \mathbf{0}$).

If the model parameters were known, the sequential control problem can be formulated as a constrained nonlinear program (NLP). We will use the solution to such NLP as a benchmark to test the dual control algorithm of previous sections (the parameters remain unknown to the dual control algorithm). The NLP formulation is to minimize $d_1^2 + d_2^2$, where $d_i = \tau_i - \boldsymbol{\theta}_i'\boldsymbol{x}$, subject to $-1 \leq u_i \leq 1$, $i = 1, 2, 3$. The solution to this quadratic programming problem can be easily obtained using a NLP solver such as LINGO (2008). The optimal solution found using LINGO is $\boldsymbol{u}^* = (u_1, u_2, u_3)' = (0.822, 0.293, 1.000)$.

The optimization of the cost to go $\widehat{J}_{k,N}$ from $t = k, ..., N$ subject to the constraints $-1 \leq u_i \leq 1$, $i = 1, 2, 3$ was carried out for $N = 20$ periods. A Matlab program (`DualControl.m`) was written to implement the dual control algorithm described above and was used to obtain all the examples in this paper (see Appendix). The program utilizes Matlab's `fmincon` procedure solve the optimization problems at each step subject to the constraints $\widehat{y}_1 \geq 3100, \widehat{y}_2 \leq 550$ which were observed to help getting better solutions in the

first few periods. A set of 10 initial points selected according to a Latin Hypercube was used to start the optimizer. The optimization of $\widehat{J}_{k,N}$ is a hard NLP problem. A single $N = 20$ simulation/optimization run took an average of 20 minutes on a Pentium 3.6 Ghz PC (see Appendix for more details about the computer implementation of the algorithm). Thus, in this case, the time until getting the next solution at each step (about one minute on average) is not too long to make the algorithm unpractical. Figure 1 shows a sample simulation of this process under the actions of the dual controller described in the previous sections. A non-informative prior ($\alpha = 0.00001, N_0 = 0$) was used for all the parameters, starting from a mean prior value of zero. The controller initially probes the process to acquire

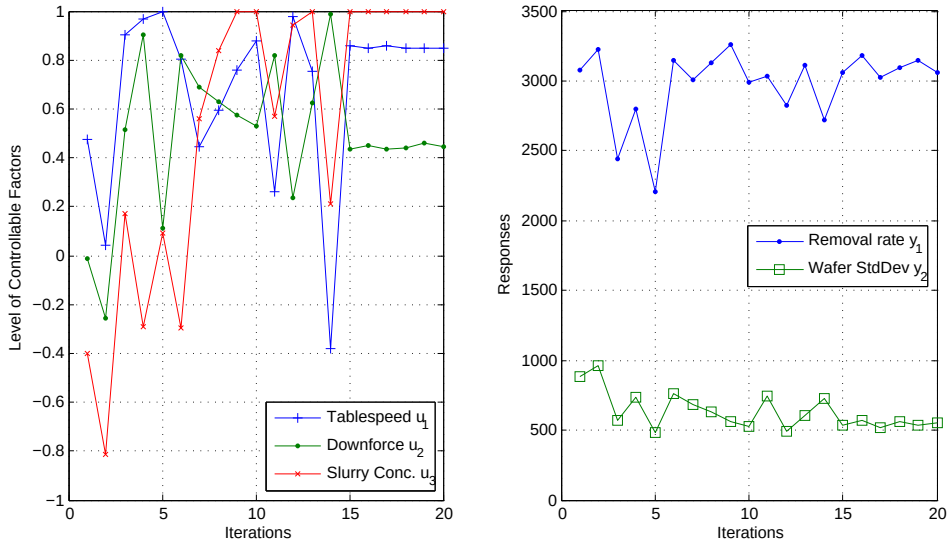

Figure 1: Sample simulation for Example 1 ($N = 20$). Left: computed controllable factors u_1, u_2, u_3. Right: observed responses y_1, y_2.

better parameter values and eventually stabilizes in a region in the controllable factor space where the responses are as close to their targets as possible. This illustrates the dual feature of the controller: current performance is sacrificed for future and (better) control actions. This is achieved without having to specify arbitrary weights that switch the objective of the controller from one of parameter estimation to one of control to target, as in Milito-like approaches (equation (1)).

One hundred independent simulation/optimizations were ran for this example using the same "true" parameters as above. Figure 2 shows a histogram of the Euclidean distances (i) $||\hat{\boldsymbol{y}}_{20} - \boldsymbol{\tau}||$, (ii) $||\boldsymbol{u}_{20} - \boldsymbol{u}^*||$ (where $\boldsymbol{u}^*$ is the NLP solution to the known parameters case), and (iii) $||\widehat{\boldsymbol{\Theta}}_{20} - \boldsymbol{\Theta}||$. As a comparison, for the NLP solution under known parameters, $||\hat{\boldsymbol{y}}_{20} - \boldsymbol{\tau}|| = 78.75$. The results indicate consistent estimation to the true parameters and convergence to the optimal solution despite the relative large errors ($\boldsymbol{\varepsilon}$). Figure 3 provides

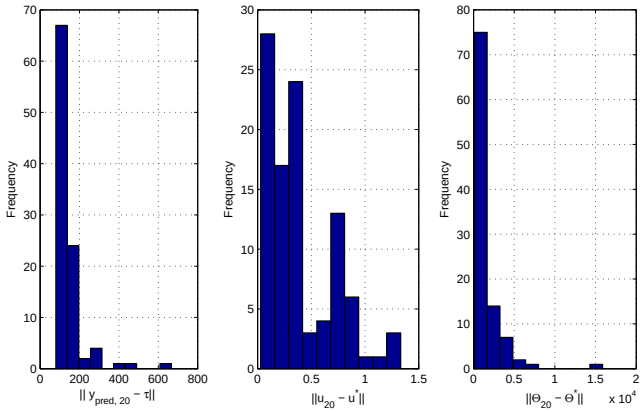

Figure 2: Results from 100 simulation/optimization cycles, Example 1. Left to right, distances from: 1) predicted responses to targets; 2) optimal settings (known parameters) to the Dual Control solution at time=20; and 3) distance from the estimated to the true parameters at t=20.

additional empirical evidence on the convergence of the Dual Control approach for this example. The value for the third controllable factor, Slurry Concentration, was set to 1 about 95% of the times. The distribution of the first factor, Tablespeed, centers at 0.8383 and the factor Downforce is centered at 0.2835. These are all very close to the NLP solution for the known-parameters case. The response values can also be contrasted to those that would be obtained

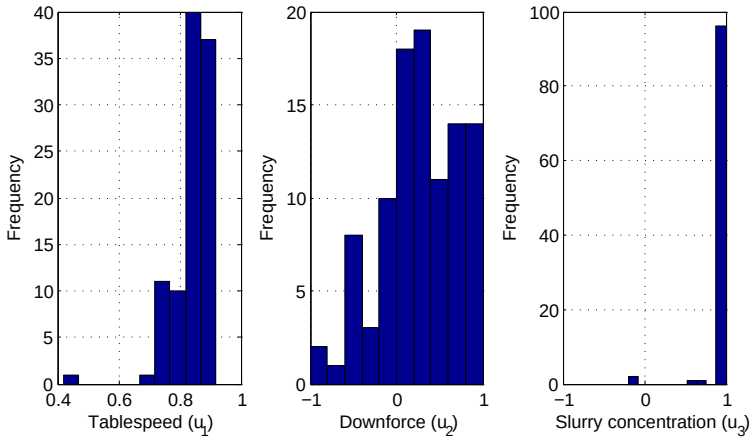

Figure 3: Solutions provided by the Dual Control algorithm, Example 1 (non-informative priors), 100 simulation/optimization cycles conducted.

if the true parameters were used in the prior and a large α value used instead to indicate a very informative prior ($\alpha = 1000$ was used). The average of the solutions over time provided by the dual control algorithm, using perfect information, was $\overline{\boldsymbol{u}}^* = (0.864, 0.3025, 1)^T$ with

corresponding standard deviations of $\hat{\sigma}(\boldsymbol{u}^*) = (0.001106, 0.0004278, 0)^T$. This provides evidence that having perfect information yields a consistent solution that is very close to the known parameters solution obtained via NLP.

Example 2. No adjustment cost, longer horizon. Consider the same process as simulated in example 1 but suppose the horizon is now $N = 100$ periods. The same non-informative priors were used. Figure 4 shows a sample simulation of this process under the actions of the dual controller for 100 iterations. Similarly to what was observed before, the

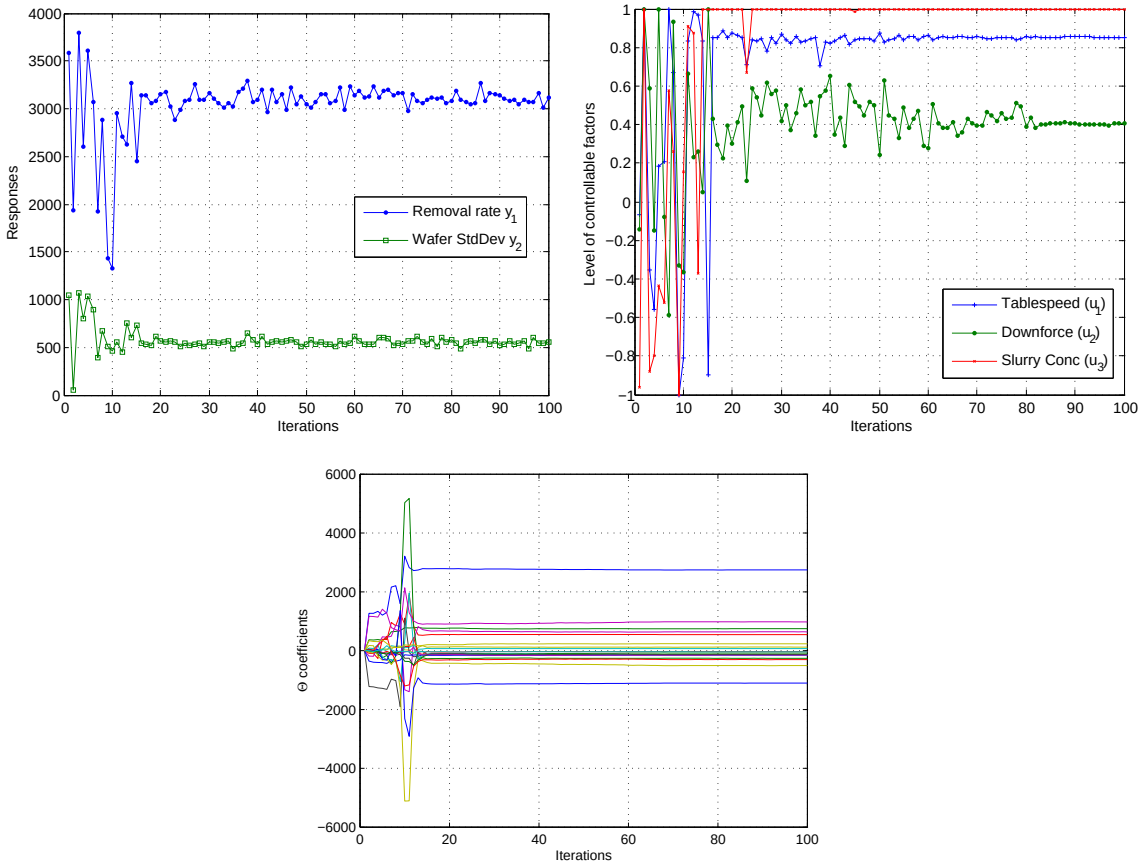

Figure 4: Sample simulation, Example 2 ($N = 100$). Top left: observed responses $(y_1, y_2)'$. Top right: computed controllable factors $(u_1, u_2, u_3)'$. Below: posterior mean of the model parameters.

controller "probes" the process initially and as soon as it has sufficient information about the process it converges to a specific solution. The time average solution for this particular simulation was $\overline{\boldsymbol{u}} = (0.853, 0.407, 1)^T$. Figure 4 also shows the estimates $\hat{\boldsymbol{\Theta}}$ over the course of this sample simulation. As the system is probed, the dual controller gathers more information about it, converging to the true parameter values.

Example 3. The effect of a high adjustment cost on a controllable factor

(hard to vary factor). In the previous examples, the cost of an adjustment was negligible compared to the cost of having the process responses off target. To investigate the effect of a non-neglible adjustment cost, the adjustment cost related to factor u_2 (downforce) was increased (Figure 5) keeping all other data as in Example 2. Here $\text{diag}(\boldsymbol{R}) = (0, r_2, 0)^T$ where r_2 was varied over the values $\{0,\ 500 \text{ and } 500 \times 10^3\}$. As expected, the adjustment variance

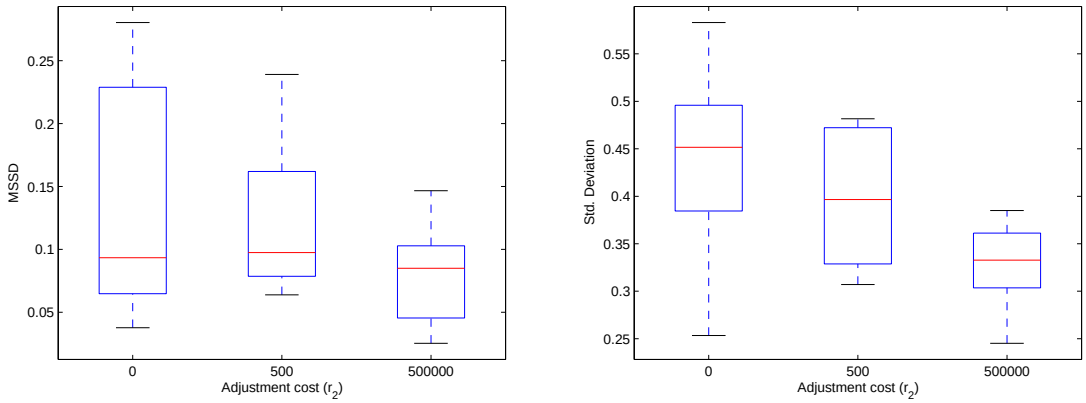

Figure 5: MSSD and SDev(u_2) for different adjustment costs r_2, Example 3 (10 simulation/optimizations).

is smaller as the adjustment cost increases. The Standard Deviation of the adjustments in the second factor and the Mean Square Successive Difference (MSSD) were computed for comparing the algorithm performance under the three different adjustment costs (r_2). The MSSD statistic (Von Neumann et al., 1941) is defined as $\text{MSSD} = \sum_{i=2}^{N}(X_{i-1} - X_i)^2/(2n-2)$ and provides an unbiased estimate of the variance. A Kruskal-Wallis test reveals that the differences between the medians of the standard deviations for each adjustment are statistically significant at a 0.05 level. It was observed that the computational time increases significantly when adjustment costs are used. Hence, results for only 10 simulation/optimization cycles are reported in Figure 5. For this example where adjustment costs were included, a single $N = 100$ simulation/optimization cycle took on average 10.2 hours on a Pentium 3.6GHz running our Matlab code (see Appendix). Naturally, giving the shorter horizons m for the later time periods, most of the computing effort occurs at the initial runs, when the planning horizon is large.

Example 4. Correlated responses. In this example, the error covariance matrix used in Example 1 was modified to reflect correlation between the two responses. The correlation ρ between the responses was set at values -0.89, -0.125, 0.125 and 0.89. Table 1 summarizes the estimated time average and standard deviations of the $N = 20$ solutions found for 100 simulations under each correlation scenario. As expected, the algorithm is more consistent when more information about the process becomes available. The standard

ρ	$\hat{E}(u_1)$	$\hat{\sigma}(u_1)$	$\hat{E}(u_2)$	$\hat{\sigma}(u_2)$	$\hat{E}(u_3)$	$\hat{\sigma}(u_3)$
-0.889	0.8415	0.0509	0.2581	0.4259	0.9993	0.0068
-0.125	0.8358	0.0667	0.2769	0.4306	0.9651	0.1952
0.000	0.8383	0.0655	0.2835	0.4733	0.9721	0.1705
0.125	0.8228	0.0960	0.2499	0.4622	0.9684	0.1827
0.889	0.8547	0.0369	0.3243	0.2488	1.0000	0.0002

Table 1: Summary statistics for solutions obtained under different correlation between errors, Example 4 (100 simulation/ optimizations, $N = 20$). Known-parameters optimal solution is $\boldsymbol{u}' = (0.822, 0.293, 1.000)$. $\rho = \pm 0.889$ corresponds to $\text{Cov}(\varepsilon_1, \varepsilon_2) = \pm 40^2$ and $\rho = \pm 0.125$ corresponds to $\text{Cov}(\varepsilon_1, \varepsilon_2) = \pm 15^2$. As the error correlation increases in magnitude, the solutions get better and more consistent.

deviation of the solutions obtained with the highly correlated errors ($\rho = \pm 0.889$) is smaller than the standard deviation of the solutions obtained in the uncorrelated case. Figure 6 shows how the algorithm converges faster in the correlated errors case. For the purposes of this example, convergence was defined heuristically as the number of periods t until $\|\boldsymbol{w}_k\| \leq 0.2, k = t, t-1, t-2$ for the first time. A one-way ANOVA reveals a significant difference in the number of iterations until convergence as the correlation between responses changes (p-value < 0.0005.) It was found that if the errors are correlated, the total solution time becomes considerably longer (see Appendix).

Example 5. A 3-response 6-factor example. Consider the fitted models reported in May et al. (1991). Four responses were fitted to model a plasma etch process; since there is an apparent error in one of their models (for Oxide Selectivity, which results in abnormal negative values) we consider only the remaining three responses. The fitted models for the remaining three responses ($p = 3$ and $q = 6$) are as follows:

$$
\begin{aligned}
y_1 &= [4498, 162, 454.25, 121.675, \text{-}350.625, \text{-}134.4, 179.5, 0, 0, \text{-}127.5, 117.3, 0, 106.25, 0, \ldots \\
&\quad \ldots, \text{-}125.4, 0, 110.625, 0, 0, 0, 0, 0, 0, 0, \text{-}82.5, \text{-}331.875, 0, 0] \cdot \boldsymbol{x} + \varepsilon_1 \\
y_2 &= [8.72625, \text{-}1.925, 2.2475, \text{-}2.1125, 6.9375, \text{-}2.67, 0, 0, 0, 0, 0, 0, \text{-}2.2125, 5.175, 0, 0, \ldots \\
&\quad \ldots, \text{-}2.625, 0, 0, 0, 0, 0, 0, 0, 0, 0, 4.48875, 0, 0] \cdot \boldsymbol{x} + \varepsilon_2 \\
y_3 &= [12.032, 0.2275, 1.245, 0.10625, \text{-}0.38625, \text{-}0.0957, \text{-}0.0325, \text{-}0.125, 0.1625, \text{-}0.2625, \ldots \\
&\quad \ldots, 0, 0.0925, 0, 0.135, 0, 0, \text{-}0.09375, 0, 0, 0, 0, 0, 0, 0.0675, 0, 0, 0.06813, 0] \cdot \boldsymbol{x} + \varepsilon_3
\end{aligned}
$$

where $\boldsymbol{x}$ is defined as in equation (3) and $\boldsymbol{V}$ is diagonal with elements 50^2, 2^2 and 0.09^2. Here, y_1 is the polysilicon etch rate in Å/min in a plasma etch modeling process, y_2 represents the etch uniformity measured, and y_3 represents the photoresist selectivity. The

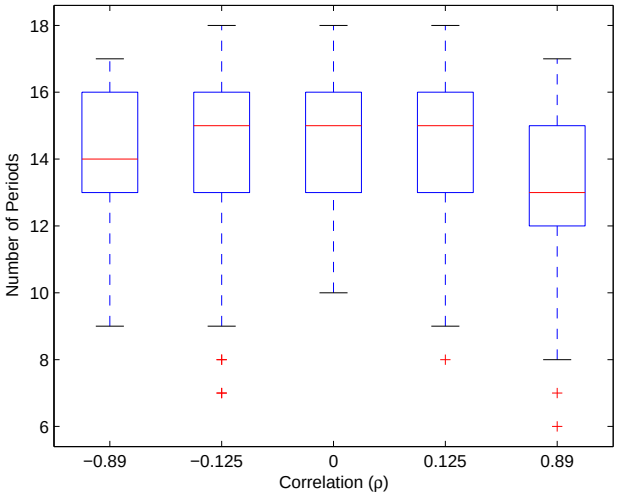

Figure 6: Number of periods until convergence (as defined in Section 6) for different correlation values, Example 4 (100 simulations/optimizations). The error covariance speeds up convergence.

controllable factors, coded in the (-1,1) range, are RF power (u_1), Pressure (u_2), Electrode Spacing (u_3), CCl_4 (u_4), Helium Flow (u_5) and Oxygen Flow (u_6). The desired targets for the responses correspond to the ideal solution where each response reaches its single optima, $\boldsymbol{\tau} = (5680, 0.7, 14.5)'$. In this example the models are assumed to be the true description of the process for simulation purposes. Similarly to what was done in Example 1, if the parameters of the model were known, the control problem can be formulated as a constrained nonlinear program. The optimal solution found using LINGO is $\boldsymbol{u}^* = (u_1, u_2, u_3, u_4, u_5, u_6)' = (1, \text{-}1, 1, \text{-}1, 1, 1)'$. The optimization of $\widehat{J}_{k,N}$ from $t = k, ..., N$ subject to the constraints $-1 \leq u_i \leq 1$, $i = 1, \ldots, 6$ was carried out for $N = 50$ periods. Matlab's `fmincon` optimization procedure was utilized to solve the optimization problems using the constraints $\widehat{y}_1 \geq 4000, \widehat{y}_2 \leq 20$ and $\widehat{y}_3 \geq 10$. A set of 5 initial points selected according to a Latin Hypercube was used to start the optimizer. Figure 7 illustrates the closeness of the Dual Control solution to the ideal vector of targets.

7 Convergence properties

Sufficient conditions for the convergence of stochastic adaptive controllers were given by Lai and Wei (1982) and further discussed by Pronzato (2000), who showed that the Milito-like dual control compound objective (eq. 1) with fixed weight ω does not satisfy the sufficient conditions of convergence. When these sufficient conditions are met, the least squares pa-

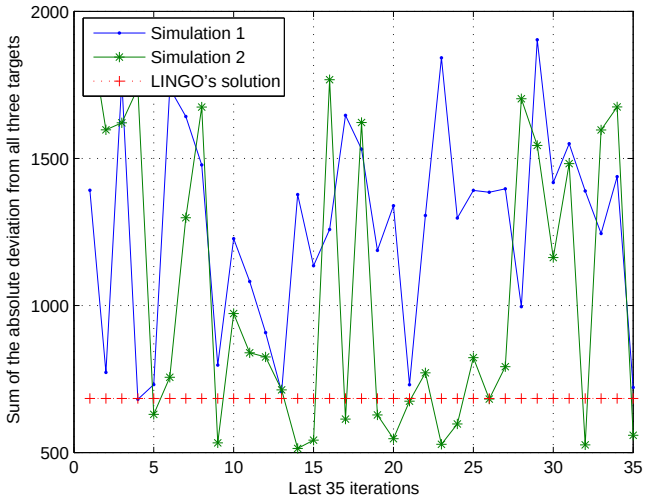

Figure 7: Performance of the proposed dual control algorithm for two sample simulations and for the known parameters solution, Example 5. Performance is measured as the sum of absolute deviations between observed responses and the target.

rameter estimates converge to the true parameters of the underlying model, assuming the model structure known. The convergence of $\boldsymbol{u}_t$ to the optimal solution of the cost function would follow if the optimization step min $J_{k,N}(\tilde{\boldsymbol{W}})$ can be solved globally. Convergence conditions for the parameters in the Bayesian case involve the eigenvalues of the $\boldsymbol{S}_{xx}$ matrix described before. If $\lambda_{\min,k}$ and $\lambda_{\max,k}$ denotes the smallest and largest eigenvalues of the $\boldsymbol{S}_{xx}$ matrix, the sufficient conditions for parameter convergence are, as $k \to \infty$:

$$\lambda_{min,k} \to \infty$$

and

$$\frac{log(\lambda_{\max,k})}{\lambda_{\min,k}} \to 1.$$

These conditions are, evidently, asymptotic conditions, not achievable in a finite horizon N. It is still of importance to determine if the proposed sequential DOE algorithm does at least approximate these conditions for larger horizons N. The conditions indicate that the smallest eigenvalue should diverge faster than the largest eigenvalue. Figure 8 shows a plot of $\lambda_{\min,k}$ and $log(\lambda_{\max,k})/\lambda_{\min,k}$ vs. k for the problem studied in example 2. After 100 observations, in agreement with the convergence conditions the smallest eigenvalue keeps increasing and the ratio in the second condition has a value close to 3.0 at $k = N$ for most of the simulations presented. The behavior depicted in the figure is representative of what we found in other numerical experiments that were conducted.

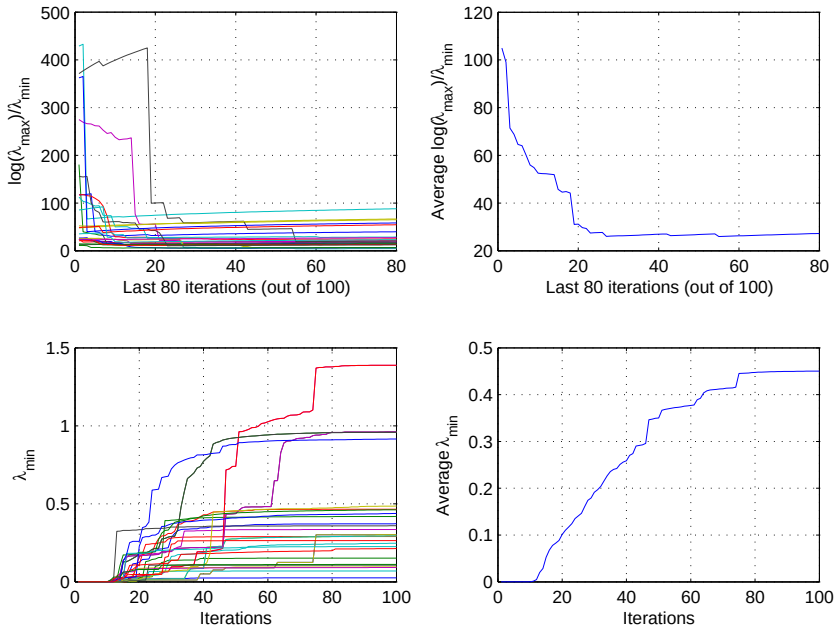

Figure 8: Convergence diagnostics $log(\lambda_{max}/\lambda_{min})$ and λ_{min} observed for the 100 simulation/optimizations conducted in Example 2. Left figures: observed values; Right figures: ensemble averages.

8 Discussion and Conclusions

A new approach for the sequential design of multiple response optimization experiments was presented. The method assumes the responses of interest can be modeled by linear statistical models, and is based on an approximation to the cost to go function based on a Matrix T predictive density. It was shown how the algorithm has dual control behavior, first probing the process and eventually converging to the optimal solution. It was shown how the algorithm satisfies the sufficient conditions for convergence as the length of the horizon increases. The proposed dual control can be justified when the cost of having the process responses running far from their targets is very expensive.

A natural alternative to the algorithm presented is to run a classical response surface experimental design, fit models, and obtain an off-line solution based on the fitted models. The solution obtained from the fitted models obtained after completion of the DOE may be quite good in general since the models may fit quite well. For fitting second order polynomial models as used in the examples of this paper, a popular DOE is a Central Composite Design (CCD), which, if the second order model form is correct, will provide excellent support for these models. However, the advantages a sequential design provides *while the DOE is run* will be lost in such approach, since no consideration is being made on the observed response

values, and the off-target cost may be unacceptably high for some runs. Likewise, the large fluctuations the controllable factors are required to have to comply with the CCD may be excessive for the "hard to vary" factors. Furthermore, fitting the responses individually will not take advantage of the correlation in the errors, if this exists. Example 4 illustrated how, as the error correlation increases, the convergence of the dual control algorithm speeds up.

To illustrate, consider the semiconductor manufacturing process described in Example 1. We simulate the models fit by Khuri (1996) as if they were the true process and evaluated the response models at each of the 17 points of a face-centered CCD (this keeps the range of the coded factors between (-1,1) and will allow for comparisons). The error variances are the same as in Example 1 but we set $\text{Cov}(\varepsilon_1, \varepsilon_2) = 40^2$. The off-line optimization problem was formulated as follows: minimize $(\hat{y}_1 - 3200)^2 + (\hat{y}_2 - 500)^2$ subject to $-1 \leq x_i \leq 1 \ \forall i$ and solved using LINGO upon completion of the CCD, which yields the solution $\boldsymbol{u}^* = (0.781, 0.3854, 1)'$. Figure 9 shows the simulated responses obtained while the CCD was run (17 runs) and for three more runs (runs 18-20) after completing the CCD at the optimal settings $\boldsymbol{u}^*$. As it can be seen, the solution obtained from the CCD is of very high quality

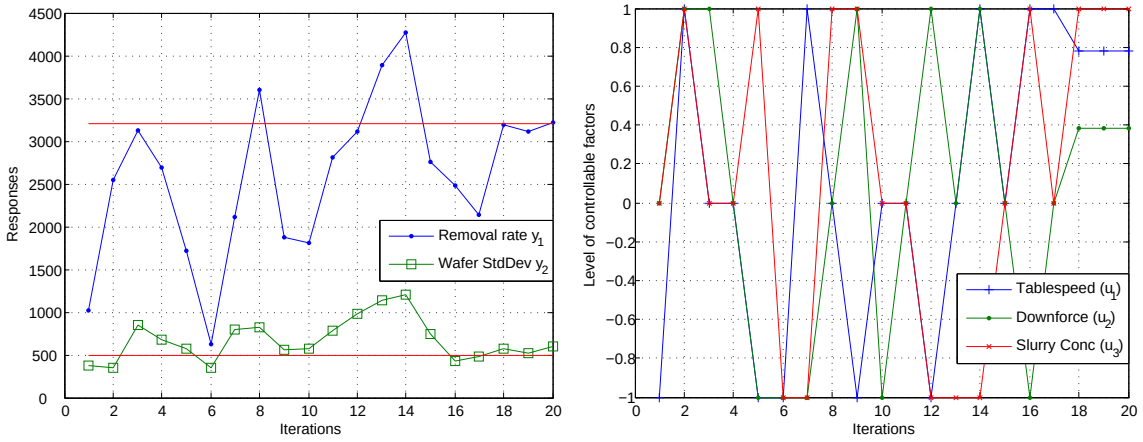

Figure 9: Left: Observed responses for all 20 periods. Horizontal lines represent the process targets. Right: Controllable factors during the face-centered CCD followed by off-line optimization at period 17, maintaining the solution for the last three iterations.

despite the correlation between the responses. This is because the model form is the same for all responses, and the least squares estimates obtained upon completion of the CCD are still the minimum variance linear unbiased estimators (Press, 1982, p. 233). The main difference with the dual control approach presented here (compare to Figure 1) is in the speed with which the optimum is reached and the more moderate fluctuations in the controllable factors during the experiments, as compared to what the CCD requires. The easiness with which "hard to vary" factors can be incorporated into the experimental problem (Example 3) is another practical feature of the dual control approach.

The proposed approach assumes a standard multivariate linear statistical model in which the form of the models (the design matrix) is the same for all responses. If this is not the case, we have what Zellner (1962) called the "Seemingly Unrelated Regression" (SUR) model. The off-line frequentist estimation of SUR models is conducted by using generalized least squares with an estimated covariance matrix. The Bayesian estimation of the SUR model has been studied by a number of authors (e.g. Percy, 1992) and requires Markov Chain Monte Carlo techniques, as there is no closed-form expression for the predictive distribution. While in principle the dual control approach presented in the present paper could be extended to the SUR case, coupling the MCMC computations with the dynamic programming optimizations will result in an extremely difficult problem from a computational point of view. If these computational obstacles can be overcome by some means for the SUR case, a model updating mechanism such as that used by Ben-Gal and Caraminis (2002) could be incorporated.

Other refinements the present paper did not explore seem more feasible to implement. For instance, the model could be expanded to incorporate noise factors, which could be simulated in the predictive density according to given distributions. Miro et al. (2004) used this approach in their (off-line) Bayesian solution to the Robust Parameter Design problem. This will increase also the computational burden (which could be reduced with multiple CPU's and a more efficient computer implementation). Finally, the possibility of developing a stopping criterion, similar to what Ben-Gal and Caramanis (2002) propose, can be studied. The idea is to stop the dual control optimizer when the marginal benefit of re-optimizing the process seems low.

References

Åström, K.J. (1970), *Introduction to Stochastic Control Theory*, Academic Press, New York, NY.

Åström, K.J. and Wittenmark, B. (1989), *Adaptive Control*, Addison-Wesley, Reading, MA.

Ben-Gal, I. and Caramanis, M. (2002), "Sequential DOE via dynamic programming", *IIE Transactions*, 34, 1087-1100.

Blum, J. (1954), "Multivariate Stochastic Approximation Methods", *Annals of Mathematical Statistics*, 25, 737-744.

Box, G.E.P., Draper, N.R. (1957), "Evolutionary operation: A method for increasing industrial productivity", *Journal of Applied Statistics*, 6, 81-101.

Box, G.E.P., Hunter, J.S., and Hunter, W.G. (2005), *Statistics for Experimenters: An Introduction to Design, Data Analysis, and Model Building*, 2nd ed., John Wiley & Sons, New York, NY.

Del Castillo, E. (2007), *Process Optimization, a Statistical Approach*, Springer, New York, NY.

Feldbaum, A. (1960), "Dual Control Theory", *Automation and Remote Control*, 21, 874-880.

Geisser, S. (1993), *Predictive Inference (Monographs on Statistics and Applied Probability)*, Chapman & Hall/CRC, New York, NY. bibitem[Gupta Nagar (2000)]GuptaNagar:2000 Gupta, A. and Nagar, D. (2000), *Matrix Variate Distributions*, Chapman & Hall / CRC, Boca Raton, FL.

Khuri, A. (1996), "Response Surface Methods for Multiresponse Experiments", *13th SEMATECH Statistical Methods Symposium (Multivariate Statistical Methods in the Semiconductor Industry)*. San Antonio, TX.

Kibria, B.M.G. (2006), "The matrix-T distribution and its applications in predictive inference", *Journal of Multivariate Analysis*, 97, 785-795.

Kiefer, J., and Wolfowtiz, J. (1952), "Stochastic estimation of the maximum of a regression function", *Annals of Mathematical Statistics*, 23, 452-466.

Kulcsar, C., Pronzato, L. & Walter, E. (1996), "Dual Control of Linearly Parameterised Models via Prediction of Posterior Densities", *European Journal of Control*, 2, 135-143.

Lai, T.L. & Wei, C.Z. (1982), "Least Squares Estimates in Stochastic Regression Models with Applications to Identification and Control of Dynamic Systems", *The Annals of Statistics*, 10, 154-166.

LINGO © 2008, Release 11.0.0.9 by LINDO Systems Inc. 1415 North Dayton Street, Chicago, IL 60622.

May, G.S., Huang, J., Spanos, C.J. (1991), "Statistical Experimental Design in Plasma Etch Modelling", *IEEE Transactions in Semiconductor Manufacturing*, 4, No. 2, 83-98.

Milito, R.; Padilla, C., Padilla, R. & Cadorin, D. (1982) "An Innovations Approach to Dual Control", *IEEE Transactions on Automatic Control*, 27, No. 1, 132-137.

Minka, T. (2001), "Bayesian Linear Regression", unpublished reference available at http://research.microsoft.com/en-us/um/people/minka/papers/.

Miro, G., Del Castillo, E., and Peterson, J.J., (2004). "A Bayesian Approach for Multiple Response Surface Optimization in the Presence of Noise Variables", *Journal of Applied Statistics*, 31 (3), pp. 251-270.

Percy, D. F. (1992), Prediction for Seemingly Unrelated Regressions, *Journal of the Royal Statistical Society Series B*, 54, pp. 243-252.

Press, S.J. (1982), *Applied Multivariate Analysis: Using Bayesian and Frequentist Methods of Inference*, 2nd Ed., Robert E. Krieger Publishing, Mallabar, FL.

Pronzato, L., Kulcsar, C., and Walter, E. (1995), "The interface between sequential experimental design and dual control", *Conf. on Control of Industrial Processes: Synthetising the statistical and Engineering Approaches, Fielder Centre, Hatfield (UK), Sept. 1995, R. Jones (ed.)*, 9-14.

Pronzato, L. (2000), "Adaptive Optimization and D-optimum Experimental Design", *The Annals of Statistics*, 28, 1743-1761.

Robbins, H. and Monro, S. (1951), "A Stochastic Approximation Method", *Annals of Mathematical Statistics*, 22, 400-407.

Spall, J.C. (1992), "Multivariate Stochastic Approximation using a simultaneous perturbation gradient approximation", *IEEE Transactions on Automatic Control*, 37, 332-341.

Spall, J.C. (2005), *Introduction to Stochastic Search and Optimization: Estimation, Simulation, and Control*, 2nd Ed., John Wiley & Sons. Hoboken, NJ.

Tse, E. & Bar-Shalom, Y. (1978), "An Actively Adaptive Control for Linear Systems with Random Parameters via the Dual Control Approach", *IEEE Transactions on Automatic Control*, 18, 109-117.

Von Neumann, J., Kent, R.H., Bellison, H.R. & Hart, B.I. (1941), "The Mean Square Successive Difference", *Annals of Mathematical Statistics*, 12, 153-162.

Wittenmark, B. (1975), "An active suboptimal dual controller for system with stochastic parameters", *Automatic Control Theory and Applications*, 3, 13-19.

Wellstead, P. & Zarrop, M. (1991), *Self-Tuning Systems: Control and Signal Processing*, Wiley, New York, NY.

Zellner, A. (1962). "An efficient method of estimating seemingly unrelated regressions and tests of aggregation bias", *Journal of the American Statistical Association*, 57, pp. 500-509.

Appendix. Computer implementation of the dual control algorithm.

The Matlab program `DualControl.m` implements the Matrix-T dual control algorithm presented in this paper and can be downloaded from

http://www2.ie.psu.edu/Castillo/research/EngineeringStatistics/software.htm

The program requires Matlab's optimization and statistics toolboxes. The `fmincon` solver was used to find the minimum of $J_{k,N}(\tilde{\boldsymbol{W}})$ (equation 9). The solver is started from a set of initial points $\boldsymbol{u}_0$ selected according to a Latin Hypersquare design, and attempts to find a minimum $\boldsymbol{u}$ satisfying a set of bounds and nonlinear constraints. These optimization steps requires most of the computing time. To illustrate, Figure 10 shows the functions consuming the largest amount of time during one of the Simulation/Optimizations conducted for Example 4 ($N = 20$, 10 initial points). The top five time-consuming routines (`qpsub, compdir, qrdelete, planerot`, and `qrinsert`) are all related to the sequential quadratic programming routines invoked by `fmincon` (e.g., `qpsub` solves the classical quadratic programming problem: Min $\frac{1}{2}\boldsymbol{x}'\boldsymbol{H}\boldsymbol{x} + \boldsymbol{f}'\boldsymbol{x}$, subject to $\boldsymbol{A}\boldsymbol{x} \leq \boldsymbol{b}$ and the other 4 routines are auxiliaries of `qpsub`). These functions accounted for 90% of the total computational time in Example 4, but the behavior was similar in all examples. The times in the figure correspond to a Pentium

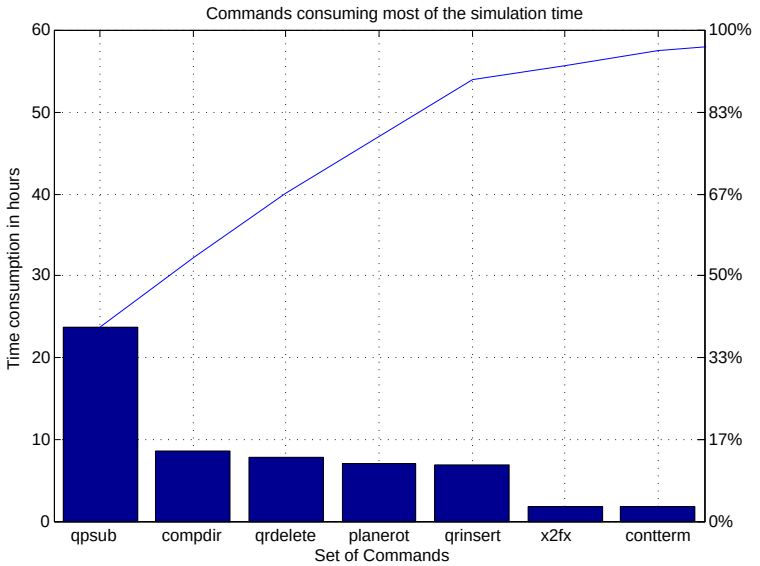

Figure 10: `DualControl.m` time-consuming functions for one simulation/optimization cycle ($N = 20$) in Example 4 (10 initial points).

3.6 Ghz single processor PC. While faster solution times can be achieved by compiling the program (recoding in C, perhaps) and/or running it in a multiple CPU PC, the above times

indicate the solution to the dual control problem is still computationally difficult. However, the times per time period indicate the overall approach is practical with modern computing power.