Policies for Simultaneous Estimation and Optimization

Miguel Sousa Lobo [1] Stephen Boyd [2]

Abstract

Policies for the joint identification and control of uncertain systems are presented. The discussion focuses on the case of a multiple input, single output linear system, with no dynamics and quadratic cost, and system parameters assumed to have a known Gaussian distribution. Extensions for multiple output, and for finite impulse response systems are straightforward. The policies proposed are heuristics, and an approximation of the optimal dynamic programming solution, that exploit convex optimization techniques. Numerical experiments are encouraging.

1 Introduction

This paper addresses the problem of controlling uncertain systems, where a policy for joint identification and control (or *dual control*) is required. While the measure of success of such a policy is by its control performance alone, it may be desirable to sacrifice some immediate control performance in order to select inputs that generate more information about the system – and thereby improve control performance in the future. If the policy is passive with respect to learning, *i.e.*, if in the selection of the inputs no attention is paid to their effect on system identification, the overall control performance can be severely degraded (in the extreme case, and for some systems, this can lead to intermittent instability, or bursting phenomena, as described by B. Anderson [1]).

This paper discusses the case of a multiple input, single output linear system, with no dynamics and quadratic cost. This choice is justified by the need to clarify concepts, and to keep expressions simple. Extensions for multiple output, and for finite impulse response systems are straightforward. The simple problem discussed here has, nevertheless, a number of industrial applications.

With the currently available convex optimization techniques, and given ever increasing processor speed and memory, convex programs can be solved in real-time at ever faster rates, which opens the way to many new control policies.

2 Problem statement

Consider a sequence of T linear input-output relations with random disturbances,

$$y_k = b^T u_k + e_k, \quad k = 1, \ldots, T. \tag{1}$$

[1]Stanford University, e-mail: mlobo@stanford.edu.

[2]Stanford University, e-mail: boyd@stanford.edu. For updates, see: http://www.stanford.edu/~boyd/groupindex.html.
Research supported in part by NSF Grant ECS-9707111, by AFOSR Grant F49620-98-1-0147, and by the Portuguese Government under Praxis XXI.

We refer to k as the time index, and to T as the horizon. Inputs $u_1, \ldots, u_T \in \mathbf{R}^n$ are to be selected, with the goal of producing outputs $y_1, \ldots, y_T \in \mathbf{R}$ with some desired property. The $e_1, \ldots, e_T \in \mathbf{R}$ are disturbances (or output noises), each with normal distribution $\mathcal{N}(0, \sigma_e^2)$, and mutually independent. The system parameters $b \in \mathbf{R}^n$ are imprecisely known. The *a priori* distribution of b is normal $\mathcal{N}(\hat{b}_0, \Sigma_0)$, and independent of the e_k. We'll assume the covariance matrix $\Sigma_0 \in \mathbf{R}^{n \times n}$ to be positive definite, and define the *a priori information matrix* as

$$\Pi_0 \triangleq \Sigma_0^{-1}.$$

The goal is to optimize a performance measure which is a function of the outputs $y_1, \ldots, y_T$. In this paper, we seek to minimize the expected value of the sum of the squares of the deviations from some desired output trajectory $y_1^{\mathrm{des}}, \ldots, y_T^{\mathrm{des}} \in \mathbf{R}$ (*i.e.*, the ℓ_2-norm of the tracking error). The full sequence $y_1^{\mathrm{des}}, \ldots, y_T^{\mathrm{des}}$ is assumed known *a priori*. In addition, we consider an additional cost term quadratic in the inputs, weighted by $\rho \geq 0$. The expected cost is then

$$\begin{aligned} \phi &\triangleq \mathbf{E}\left(\sum_{k=1}^{T} \left(y_k - y_k^{\mathrm{des}} \right)^2 + \rho\, u_k^T u_k \right) \qquad (2) \\ &= \mathbf{E}\left(\sum_{k=1}^{T} \left(b^T u_k + e_k - y_k^{\mathrm{des}} \right)^2 + \rho\, u_k^T u_k \right). \end{aligned}$$

The expectation is over the distributions of $b, e_1, \ldots, e_T$.

We define a *feasible policy* to be one where the choice of the u_k is non-anticipating in k, in the sense that it relies only on information available up to time $k-1$, in particular on $y_1, \ldots, y_{k-1}$. It may also use the *a priori* information about the distributions of b and $e_1, \ldots, e_T$, and about the full sequence of desired outputs $y_1^{\mathrm{des}}, \ldots, y_T^{\mathrm{des}}$. Formally, the problem consists in finding the functions $\psi_1, \ldots, \psi_T$, of the form

$$u_k = \psi_k(\hat{b}_0, \Pi_0, \sigma_e^2, y_1^{\mathrm{des}}, \ldots, y_T^{\mathrm{des}}, y_1, \ldots, y_{k-1}), \tag{3}$$

that minimize (2). In a feasible policy, u_k is a random variable measurable $\sigma(y_1, \ldots, y_{k-1})$. (We'll also consider a generalization, which we call *randomized feasible policy*, where u_k is measurable $\sigma(y_1, \ldots, y_{k-1}, w)$, with w some independent random variable introduced to allow for the randomization of u_k.)

Intuitively, finding the best input u_k requires solving the tradeoff between 1) choosing a u_k expected to produce an output that is close to y_k^{des} and 2) introducing perturbations in the input to improve knowledge of b and, as a consequence, obtain better performance in problems $k+1, \ldots, T$. The two goals may conflict. For instance, if a zero output is desired at some time k, the smallest expected error is obtained by selecting a zero input u_k. But a zero input is also the least informative. In terms of the overall expected cost, it may be better to select a (small) non-zero input that is more informative, in the sense that it improves the accuracy with which we can estimate b, leading to improved tracking performance in future times $k+1$ to T. This trade-off between design for estimation (or experiment design, or system identification) and optimization (or control), is the central concern of our study.

The true solution to the problem is given by a dynamic program which is very hard to solve numerically. It requires numerical integration over a high dimensional space (what has been called the "curse of dimensionality"). We propose an approximation which results in a semidefinite program, for which very effective solution methods have been developed in recent years. Prior to the development of these algorithms, the approximation we introduce might have been considered almost as complex as the original problem. For further discussion of dual control and dynamic programming, see A. Fel'dbaum [2, 3], Y. Bar-Shalom [4, 5, 6], Kumar and Varaiya [7, §6.8], and D. Bertsekas [8, §6].

The other approach discussed in this paper is heuristic in nature, and relates to some recent work on plant-friendly identification (see Genceli and Nikolaou [9], and Cooley and Lee [10]). We place this idea in a more general and productive framework, by introducing measures of input informativeness.

Results for this problem translate directly to the *receeding horizon* case, where after the application of the first input u_1 the problem is extended to include consideration of y_{T+1}^{des} (so that the horizon T remains constant). Note, however, that receeding horizon control is but a heuristic for the infinite horizon problem, which we do not address in this paper. For references on model predictive predictive control see, *e.g.*, R. Bitmead et al. [11].

2.1 Conditional distribution of b

This section succinctly presents, for later use, standard results on the conditional distribution of b given the outputs $y_1, \ldots, y_k$ (see, *e.g.*, L. Ljung [12]). Define

$$U_k \triangleq \begin{bmatrix} u_1^T \\ \vdots \\ u_k^T \end{bmatrix}, \quad Y_k \triangleq \begin{bmatrix} y_1 \\ \vdots \\ y_k \end{bmatrix}, \quad Y_k^{\text{des}} \triangleq \begin{bmatrix} y_1^{\text{des}} \\ \vdots \\ y_k^{\text{des}} \end{bmatrix}.$$

The conditional distribution of b given U_k and Y_k is normal $\mathcal{N}(\hat{b}_k, \Sigma_k) = \mathcal{N}(\hat{b}_k, \Pi_k^{-1})$, with

$$\begin{aligned} \Pi_k &= \Pi_0 + \sigma_e^{-2} U_k^T U_k, \\ \hat{b}_k &= \Pi_k^{-1}\left(\Pi_0 \hat{b}_0 + \sigma_e^{-2} U_k^T Y_k\right). \end{aligned}$$

Equivalent, recursive formulas are

$$\begin{aligned} \Pi_{k+1} &= \Pi_k + \sigma_e^{-2} u_{k+1} u_{k+1}^T, \\ \hat{b}_{k+1} &= \hat{b}_k + \sigma_e^{-2} \Pi_{k+1}^{-1} u_{k+1}\left(y_{k+1} - \hat{b}_k^T u_{k+1}\right). \end{aligned}$$

Note that, since all distributions are assumed normal, $\hat{b}_k$ and Π_k^{-1} are sufficient statistics and summarize all information available about b. They can be interpreted as a system state. We'll also use the notation $\tilde{b}_k = b - \hat{b}_k$.

3 Passive learning (adaptive control)

3.1 Certainty equivalent policy

This section addresses a suboptimal feasible policy, which is the equivalent of *adaptive control* applied to our problem. Consider the conditional distribution, described by $\hat{b}_{k-1}$ and Π_{k-1}, and updated as described in §2.1. At each time index k, the input u_k is chosen as if b was precisely known, and equal to $\hat{b}_{k-1}$. That is, the input is selected as if $\Sigma_{k-1} = 0$ and, in this sense, we call it a *certainty equivalent* policy. In the selection of the input at each time step k, we use $\hat{b}_{k-1}$ which depends on $y_1, \ldots, y_{k-1}$ ($\hat{b}_{k-1}$ also depends on $u_1, \ldots, u_{k-1}$, which are in turn functions of $y_1, \ldots, y_{k-2}$). If it were true that $\Sigma_{k-1} = 0$, the expected cost conditional on Y_{k-1} would be

$$\begin{aligned} \phi \;=\; & \textstyle\sum_{l=1}^{k-1} (y_l - y_l^{\text{des}})^2 + \sigma_e^2 + (\hat{b}_{k-1}^T u_k - y_k^{\text{des}})^2 + \rho\, u_k^T u_k \\ & + \textstyle\sum_{l=k+1}^{T}\left(\sigma_e^2 + \mathbf{E}\left((\hat{b}_{k-1}^T u_l - y_l^{\text{des}})^2 + \rho\, u_l^T u_l\right)\right). \end{aligned}$$

Differentiating with respect to u_k and equating to zero, we obtain the desired policy,

$$u_k \;=\; (\hat{b}_{k-1}\hat{b}_{k-1}^T + \rho I)^{-1} \hat{b}_{k-1} y_k^{\text{des}} \;=\; \frac{1}{\|\hat{b}_{k-1}\|^2 + \rho} \hat{b}_{k-1} y_k^{\text{des}},$$

where we used the matrix inversion lemma for a rank one update (and the fact that, if $\Sigma_{k-1} = 0$, $u_{k+1}, \ldots, u_T$ are independent of u_k). With this policy, the expected cost (using the true value of $\Sigma_{k-1} \neq 0$) is

$$\phi = T\sigma_e^2 + \mathbf{E}\left(\sum_{k=1}^{T} \frac{(y_k^{\text{des}})^2}{\rho + \|\hat{b}_{k-1}\|^2}\left(\rho \;+\; \frac{\hat{b}_{k-1}^T \Pi_{k-1}^{-1} \hat{b}_{k-1}}{\rho + \|\hat{b}_{k-1}\|^2}\right)\right).$$

Note that for $k = 1, \ldots, T-1$, $\hat{b}_k$ and Π_k are random variables, because they are functions of $y_1, \ldots, y_{k-1}$.

In this policy, $\hat{b}_{k-1}$ is used as an estimate of b. The accuracy of this estimate improves at each time index, due to the information gained from successive outputs (summarized in the updating of Π_k and $\hat{b}_k$). From the last equation we see that small $\Pi_0, \ldots, \Pi_{T-1}$ yield a large expected cost (where small here may be taken to mean, *e.g.*, a small $\lambda_{\min}$). Nevertheless, in the selection of $u_1, \ldots, u_{T-1}$, no effort is made to make the Π_k large (note, from §2.1, that Π_k is quadratic in $u_1, \ldots, u_k$). The inputs are designed without regard for their effect on the estimation procedure, warranting the term *passive learning.*

3.2 Regularized policy

Consider now another passive learning policy, where instead of using the certainty equivalent approximation at time k, the conditional distribution of b given $y_1, \ldots, y_{k-1}$ is taken into account. The input u_k is selected to minimize only the immediate expected cost given the available information

$$\begin{aligned} &\mathbf{E}\left(\left(y_k - y_k^{\text{des}}\right)^2 + \rho u_k u_k^T \,\middle|\, Y_{k-1}\right) = \\ &\quad = u_k^T \Pi_{k-1}^{-1} u_k + \left(\hat{b}_{k-1}^T u_k - y_k^{\text{des}}\right)^2 + \sigma_e^2 + \rho u_k u_k^T. \end{aligned}$$

The minimizing input, obtained by differentiating and equating to zero, is

$$u_k \;=\; \left(\Pi_{k-1}^{-1} + \hat{b}_{k-1}\hat{b}_{k-1}^T + \rho I\right)^{-1} \hat{b}_{k-1} y_k^{\text{des}}.$$

Note that Π_{k-1}^{-1} can be seen as a regularization term. In some sense, it adds a measure of caution to account for the uncertainty in the estimate of b. As $\|\Pi_{k-1}\| \to 0$, the optimal input goes to zero. The expected cost, for $\rho = 0$, simplifies to

$$\phi \;=\; T\sigma_e^2 + \sum_{k=1}^{T} \mathbf{E}\left(\frac{(y_k^{\rm des})^2}{1+\hat{b}_{k-1}^T \Pi_{k-1}\hat{b}_{k-1}}\right).$$

As $\|\Pi_{k-1}\| \to 0$, the minimum expected cost approaches an upper bound, which is the cost of selecting a zero input.

Again, small $\Pi_0, \ldots, \Pi_{T-1}$ yield a large expected cost. The policy is suboptimal because the selection of u_k does not take into account its effect on $\Pi_k, \ldots, \Pi_{T-1}$ (which are quadratic in u_k). This is, in effect, a *greedy* policy: At each time index, u_k is selected to minimize the immediate expected cost, $\mathbf{E}((y_k - y_k^{\rm des})^2)$, without regard for future costs. As in §3.1, there is no design for estimation, in the sense that the benefits to be gained from selecting inputs that make the Π_k large are not considered.

3.3 Derivative of the expected cost with respect to information

An invaluable and generally overlooked fact is that, for many regularized or robust policies, the derivative of the cost with respect to the information matrix is easily computed. For the regularized passive learning policy with $\rho = 0$, we have that

$$\frac{d}{d\,\Pi_0}\left(\sigma_e^2 + \frac{(y_1^{\rm des})^2}{1+\hat{b}_0^T\Pi_0\hat{b}_0}\right) \;=\; -\frac{(y_1^{\rm des})^2}{\left(1+\hat{b}_0^T\Pi_0\hat{b}_0\right)^2}\,\hat{b}_0\hat{b}_0^T.$$

4 Persistency of excitation, dithering, and maximally informative inputs

We have seen that the cost incurred at time k may be large if $\lambda_{\min}(\Pi_k)$ is small. In other words, if the covariance $\Sigma_k = \Pi_k^{-1}$ is large, $\hat{b}_k$ is an unreliable estimate of the system parameters b, and this leads to poor performance. The most immediate solution is to ensure that some measure of information, such as $\lambda_{\min}(\Pi_k) = \lambda_{\min}(\Pi_0 + \sigma_e^{-2}U_k^TU_k)$, is large. This can be translated into a requirement that U_k be well-conditioned, which is what is usually meant by *persistency of excitation.* Informally, we want the u_k to span the whole input space.

4.1 Dithering

Several solutions have been proposed to satisfy the persistency of excitation requirement. *Dithering* is a randomized feasible policy that consists of adding to the inputs some white noise (*i.e.*, normal, zero mean and independent random terms). This can work well since, given a high enough noise level, Π_k will be well-conditioned with high probability. Although it has the advantage of very simple implementation, dithering is obviously a sub-optimal policy, and the selection of a good noise level can be problematic.

Example

Consider the problem described by

$$n=2,\quad \hat{b}_0 = \begin{bmatrix} 0 \\ 1 \end{bmatrix},\quad \Pi_0 = \begin{bmatrix} 6 & 2 \\ 2 & 10 \end{bmatrix},\quad \sigma_e^2 = 0.1,$$
$$T=10,\quad Y_{10}^{\rm des} = [\,0\;\;0\;\;0\;\;0\;\;0\;\;10\;\;10\;\;10\;\;10\;\;10\,]^T.$$

We implement a dithered policy based on regularized passive learning (§3.2). Figure 1 plots the expected cost as the variance of the random terms (the *input noise level*) ranges from 1.0×10^{-9} to 10. These values were obtained by Monte Carlo simulation, with 1000 runs at each input noise level (the corresponding error bars are also plotted). Note that this may seem counterintuitive: The performance of the policy is improved by adding independent noise to the inputs. As the noise level goes to zero, we approach the non-dithered regularized passive learning policy, which has an average cost of 9.3 ($\pm$0.43). At the optimal input noise level (selected *a posteriori*), the average cost is 4.0 ($\pm$0.19). Of course, any practical dithering policy must select the input noise level *a priori*, which may be difficult.

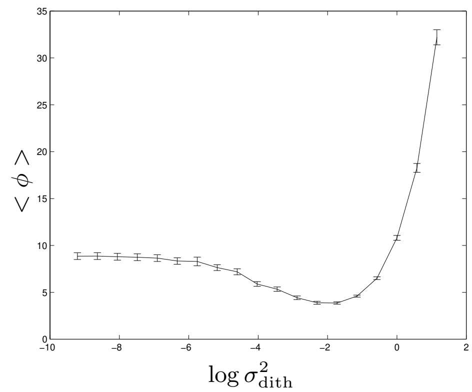

Figure 1: Expected cost as a function of dithering level.

4.2 Measuring and valuing information

A more thoughtful approach is to select a perturbation that, for a given level of control disturbance, maximizes the information gathered. For this purpose, we need a measure of information. A list of possible measures, using the naming convention from experiment design, is

- E-optimal: $\quad {\lambda_{\max}}^{-1}(\Sigma_k) \;=\; \lambda_{\min}(\Pi_k)$
- D-optimal: $\quad \log\det \Pi_k$
- A-optimal: $\quad -\,\mathbf{Tr}\,(\Sigma_k) \;=\; -\,\mathbf{Tr}\,\left(\Pi_k^{-1}\right)$

The E-optimal and A-optimal measures may be scaled to account for the derivative of the expected cost with respect to information (§3.3), while the D-optimal measure is invariant with scaling. For a numerically effective heuristic, we would like a measure that is concave in the inputs. These measures are concave-quadratic, and linearizing Π_k in U_k makes them concave. This linearization can be expected to work well if the perturbations introduced for the purpose of identification are small. We return to this point in §5.5.

4.3 Maximally informative inputs

One approach consists of expressing explicitly the trade-off between control and information, by adding to the objective function an extra term valuing information. A very simple example of such a policy is to select the u_1 that minimizes

$$u_1^T\Pi_0^{-1}u_1 + (\hat{b}_0^Tu_1 - y_1^{\rm des})^2 + \rho u_1 u_1^T - \gamma\,\lambda_{\min}(\Pi_1),$$

and likewise for $u_2, \ldots, u_T$, with the appropriate updating of Π_k and $\hat{b}_k$. The extra term makes this policy, in part, an experiment design problem. As with dithering, a perturbation will be introduced in the input, what we might now call an "intelligent noise". The factor $\gamma > 0$ weighs the trade-off between identification and control. Selecting γ presents the same difficulties as for selecting the dithering level.

An alternative approach is what has been termed *plant-friendly identification*. Although it is essentially a solution to a different problem, plant-friendly identification can be used as a heuristic for simultaneous estimation and control. The idea is to select the maximally informative input from within the set of inputs that keep some measure of the tracking error within a bound. For a simple example, we use a constraint on the absolute tracking error (specified by $M \in \mathbf{R}$). The policy is defined by the program

$$\begin{array}{ll} \text{maximize} & \lambda_{\min}(\Pi_1) \\ \text{subject to} & |\hat{b}_0^T u_1 - y_1^{\text{des}}| \le M. \end{array}$$

The bound M can be seen as the trade-off factor, with a role similar to γ in the previous problem. However, M is a more "physically meaningful" number, and should be easier to select in practical applications. The constraint on performance used here disregards the uncertainty in b. A robust constraint can be used in its place (such a constraint is convex in the inputs – in fact, it is a second-order cone constraint, see S. Boyd et al. [13]).

Both these problems are convex if we linearize Π_1 in u_1, in which case they are readily solved. If this procedure – linearization followed by the solution of a convex program – is iterated, a (local) minimum of the non-convex problem will be reached.

Note that these heuristics do not use the future desired outputs y_k^{des}, which we assumed known. This is part of the suboptimal nature of the heuristics, and has the effect of reducing the sensitivity of their performance with respect to the future trajectory. This reduced sensitivity may be a positive feature in applications where the future trajectory is not fully certain.

5 Optimal policy, dynamic program and approximation

These heuristic approaches still leave us with some questions, in particular about 1) what measure of information to use, and 2) how to decide on the inevitable trade-off between the informativeness of u_k and the output error expected to result from its application. Roughly speaking, the answer to the second question is that the trade-off should be such that the current loss in tracking performance (incurred for the sake of informativeness) equals the total expected future gains in tracking performance (due to improved information about the system). This, in turn, leads to an answer for the first question: The information measure should be such that it captures the expected future gain in tracking performance.

The true solution to the problem is given by a dynamic program, of which we will outline the derivation. This dynamic program is, however, hard to solve. We propose an approximation which results in a semidefinite program.

5.1 Optimal policy for $T = 1$

We assume, from here on, $\rho = 0$. Consider the simplest case, where $T = 1$. An input u_1 is to be selected so as to minimize the expected cost

$$\begin{aligned} \phi_{T=1} &\triangleq \mathbf{E}_{b,e_1}\left(\left(y_1 - y_1^{\text{des}}\right)^2\right) \\ &= \mathbf{E}_{b,e_1}\left(\left(b^T u_1 + e_1 - y_1^{\text{des}}\right)^2\right) \\ &= \mathbf{E}_{b,e_1}\left(\left(\underbrace{\tilde{b}_0^T u_1}_{1} + \underbrace{(\hat{b}_0^T u_1 - y_1^{\text{des}})}_{2} + \underbrace{e_1}_{3}\right)^2\right) \\ &= \underbrace{u_1^T \Pi_0^{-1} u_1}_{1} + \underbrace{(\hat{b}_0^T u_1 - y_1^{\text{des}})^2}_{2} + \underbrace{\sigma_e^2}_{3}. \end{aligned}$$

The three terms marked can be interpreted as 1) the cost due to inaccuracy in the estimate of b, 2) the cost due to deviation from the certainty equivalence policy, and 3) the cost due to output noise. The input that minimizes this function, obtained by differentiating and equating to zero, is

$$u_1 = \psi_1(\hat{b}_0, \Pi_0, \sigma_e^2, y_1^{\text{des}}) = \left(\Pi_0^{-1} + \hat{b}_0 \hat{b}_0^T\right)^{-1} \hat{b}_0 y_1^{\text{des}}.$$

Note that Π_0^{-1} can be seen as a regularization term. As $\|\Pi_0\|$ becomes small, the optimal input goes to zero. The minimum expected cost is

$$\phi_{T=1}^*(\hat{b}_0, \Pi_0, \sigma_e^2, y_1^{\text{des}}) = \sigma_e^2 + \frac{(y_1^{\text{des}})^2}{1 + \hat{b}_0^T \Pi_0 \hat{b}_0},$$

where we used the matrix inversion lemma for a rank one update. Note that $\phi_{T=1}^*$ is convex in Σ_0 and concave in Π_0. For small $\|\Pi_0\|$, the minimum expected cost approaches an upper bound, which is the cost of selecting a zero input.

5.2 Optimal policy for $T = 2$

For $T = 2$, the expected cost is

$$\begin{aligned} \phi_{T=2} &\triangleq \mathbf{E}_{b,e_1,e_2}\left(\left(y_1 - y_1^{\text{des}}\right)^2 + \left(y_2 - y_2^{\text{des}}\right)^2\right) \\ &= \mathbf{E}_{b,e_1,e_2}\Big((\tilde{b}_0^T u_1 + (\hat{b}_0^T u_1 - y_1^{\text{des}}) + e_1)^2 \\ &\quad + (\tilde{b}_1^T u_2 + (\hat{b}_1^T u_2 - y_2^{\text{des}}) + e_2)^2\Big) \\ &= \mathbf{E}_{b,e_1}\left((\tilde{b}_0^T u_1 + (\hat{b}_0^T u_1 - y_1^{\text{des}}) + e_1)^2\right) \\ &\quad + \mathbf{E}_{y_1}\left(\mathbf{E}_{b,e_1,e_2}\left((\tilde{b}_1^T u_2 + (\hat{b}_1^T u_2 - y_2^{\text{des}}) + e_2)^2 \middle| y_1\right)\right) \\ &= u_1^T \Pi_0^{-1} u_1 + (\hat{b}_0^T u_1 - y_1^{\text{des}})^2 + \sigma_e^2 \\ &\quad + \mathbf{E}_{y_1}\left(u_2^T \Pi_1^{-1} u_2 + (\hat{b}_1^T u_2 - y_2^{\text{des}})^2 + \sigma_e^2\right), \end{aligned} \tag{4}$$

where (from §2.1)

$$\Pi_1 = \Pi_0 + \sigma_e^{-2} u_1 u_1^T, \qquad \hat{b}_1 = \Pi_1^{-1}\left(\Pi_0 \hat{b}_0 + \sigma_e^{-2} u_1 y_1\right).$$

We used the tower property of conditional expectation, and the fact that, if y_1 is given, then $\hat{b}_1$ and u_2 are constants and $\tilde{b}_1$ has zero mean and covariance Π_1^{-1}. Also, it is trivial to see that $\tilde{b}_0$ and e_1 are independent, and $\tilde{b}_1$ and e_2 are independent.

$\phi_{T=2}$ is to be minimized over $u_1 = \psi_1$ and $u_2 = \psi_2$, with ψ_2 a function of y_1 and u_1 (both ψ_1 and ψ_2 are also functions of $\hat{b}_0$, Π_0, σ_e^2, y_1^{des} and y_2^{des}, but for clarity these parameters will be omitted). The minimum of $\phi_{T=2}$ can be found by minimizing first over ψ_2 (*i.e.*, finding the minimizing second input u_2 as a function of the first input u_1 and output y_1). To find the minimum of (4) we will need to compute

$$\begin{aligned} \inf_{\psi_2(\cdot,\cdot)} \mathbf{E}_{y_1} \Big(& \psi_2(u_1,y_1)^T \Pi_1^{-1} \psi_2(u_1,y_1) \\ & + \left(\hat{b}_1^T \psi_2(u_1,y_1) - y_2^{\text{des}}\right)^2 + \sigma_e^2 \Big) = \\ = \; & \mathbf{E}_{y_1} \Big(\inf_{\psi_2(\cdot,\cdot)} \Big(\psi_2(u_1,y_1)^T \Pi_1^{-1} \psi_2(u_1,y_1) \\ & + \left(\hat{b}_1^T \psi_2(\psi_1,y_1) - y_2^{\text{des}}\right)^2 + \sigma_e^2 \Big)\Big) \\ = \; & \mathbf{E}_{y_1} \left(\phi^*_{T=1}(\hat{b}_1, \Pi_1, \sigma_e^2, y_2^{\text{des}}) \right) \\ = \; & \sigma_e^2 + (y_2^{\text{des}})^2 \mathbf{E}_{y_1} \left(\frac{1}{1+\hat{b}_1^T \Pi_1 \hat{b}_1} \right). \end{aligned} \tag{5}$$

We conclude that the minimum expected cost is

$$\begin{aligned} \phi^*_{T=2}(\hat{b}_0, \Pi_0, \sigma_e^2, y_1^{\text{des}}) \; = \; & \inf_{u_1} \Big(u_1^T \Pi_0^{-1} u_1 \\ + (u_1^T \hat{b}_0 - y_1^{\text{des}})^2 + \sigma_e^2 + & \mathbf{E}_{y_1} \left(\phi^*_{T=1}(\hat{b}_1, \Pi_1, \sigma_e^2, y_2^{\text{des}}) \right) \Big). \end{aligned}$$

Note that we have just derived Bellman's *principle of optimality* from first principles for this particular problem. The solution requires computing an integral of the form

$$\mathbf{E}_X \left(\frac{1}{a_0 X^2 + a_1 X + a_2} \right), \qquad X \sim \mathcal{N}(0, \sigma^2).$$

where

$$\begin{aligned} & a_0 \triangleq \sigma_e^{-4} u_1^T \Pi_1 u_1, \quad a_1 \triangleq 2\hat{b}_0^T u_1 \sigma_e^{-2}, \quad a_2 \triangleq 1 + \hat{b}_0^T \Pi_1 \hat{b}_0, \\ & X \triangleq \tilde{b}_0^T u_1 + e_1 \; \sim \; \mathcal{N}(0, \sigma^2), \qquad \sigma^2 \triangleq u_1^T \Pi_0^{-1} u_1 + \sigma_e^2. \end{aligned}$$

The denominator polynomial can be shown to be positive for all X. If an iterative optimization procedure is to be used, this expectation must be evaluated numerically at each iteration. Alternatively, we will propose using a simple approximation.

Example

Consider the previous example (in §4.1), but with a shorter horizon. In particular,

$$T = 2, \qquad Y_2^{\text{des}} = [\; 0 \;\; 10 \;]^T,$$

and $n, \hat{b}_0, \Pi_0, \sigma_e^2$ as before. For a given u_1, the expected cost ϕ is computed assuming that u_2 is selected optimally at $k = 2$. The expectation in (5) is evaluated by numerical integration. Ranging over values for the two entries of u_1, this produces Figure 2. The optimum is achieved at $u_1^* = \left[\; 0.200 \quad 0.998 \;\right]^T$, for which the expected cost is $\phi^* = 2.568$. This is to be compared with the standard procedure of minimizing the expected square error at each time step (*i.e.*, the regularized passive learning policy), which yields $u_1^* = \left[\; 0 \quad 0 \;\right]^T$, and $\phi = 9.111$.

5.3 Approximate solution for $T = 2$

Consider the approximation

$$\mathbf{E}_X \left(\frac{1}{a_0 X^2 + a_1 X + a_2} \right) \approx \frac{1}{a_2}.$$

With this approximation, the problem becomes that of minimizing

$$u_1^T \Pi_0^{-1} u_1 + (u_1^T \hat{b}_0 - y_1^{\text{des}})^2 + 2\sigma_e^2 + \frac{(y_2^{\text{des}})^2}{1 + \hat{b}_0^T \Pi_1 \hat{b}_0}$$

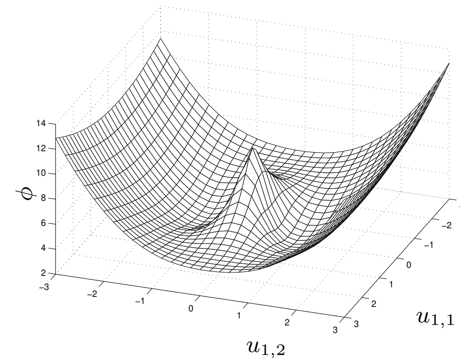

Figure 2: Expected cost as a function of first and second entry of u_1 (with u_2 optimal).

over $u_1 \in \mathbf{R}^n$. "Undoing" the minimization over u_2, we see that this is equivalent to minimizing

$$\begin{aligned} f_{T=2} \quad \triangleq \quad & u_1^T \Pi_0^{-1} u_1 + u_2^T (\Pi_0 + \sigma_e^{-2} u_1 u_1^T)^{-1} u_2 \\ & + (u_1^T \hat{b}_0 - y_1^{\text{des}})^2 + (u_2^T \hat{b}_0 - y_2^{\text{des}})^2 + 2\sigma_e^2 \end{aligned}$$

over $u_1, u_2 \in \mathbf{R}^n$. This approximation is equivalent to making the approximation $\hat{b}_1 \approx \hat{b}_0$ in (4), which will be the motivation for an extension of the approximation for any $T > 2$. We will take $\hat{b}_k \approx \hat{b}_0, k = 1, \ldots, T-1$, in the equivalent expression for the expected cost.

An intuitive description of this approximation is as follows. First, note that the *a priori* distribution of b can be described by the ellipsoid $\|(x - \hat{b}_0)^T \Pi_0 (x - \hat{b}_0)\| \le 1$ (the maximum volume set with a given probability). Likewise, the conditional distribution of b given y_1 can be described by the ellipsoid $\|(x - \hat{b}_1)^T \Pi_1 (x - \hat{b}_1)\| \le 1$. The total cost will depend on both the centers $(\hat{b}_0, \hat{b}_1)$ and the volumes (defined by Π_0, Π_1) of the two ellipsoids. From one time index to the next, with the added knowledge of y_1, the center and volume of the ellipsoid change (see Figure 3). The center changes randomly, and this is the term that introduces increased complexity in the dynamic program (as a side note, this random change has a zero mean normal distribution that depends on the inputs, and is easily computed). On the other hand, the volume changes in a deterministic fashion. Given the inputs, this change in volume can be precisely predicted. With the approximation described, we are assuming that the change in volume is more important in determining the cost than the change in center, *i.e.*, we assume that the cost is much less sensitive to the mean of the distribution than to its covariance. This is reasonable for systems that are not overdetermined, which includes our problem.

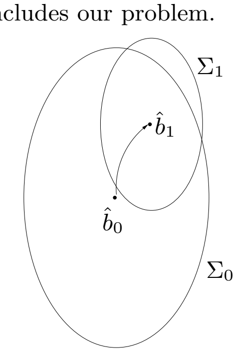

Figure 3: Changes in the conditional distribution of b.

Example

With the same example as in §5.2, for a given u_1 we compute $f_{T=2}$. Again we assume that u_2 is selected optimally. Ranging over values for the two entries of u_1, this produces Figure 4. The minimum of the approximate objective function $f_{T=2}$ is achieved at $u_1 = \begin{bmatrix} 0.184 & 0.918 \end{bmatrix}^T$. The approximate expected cost at this point is $f_{T=2} = 1.997$, and the true expected cost is $\phi_{T=2} = 2.592$. The performance degradation relative to the optimal policy is 0.9% with the approximation, as compared to 255% with the regularized passive learning policy. Figure 5 plots the approximation error as a function of u_1. Note the small error in the region where the optimum is located, which seems to be a general feature of this approximation.

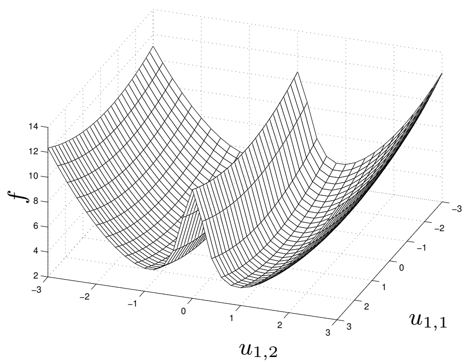

Figure 4: Approximation of the expected cost.

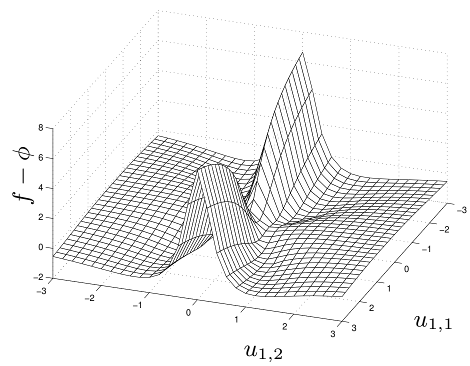

Figure 5: Approximation error.

5.4 Optimal policy and approximation for $T > 2$

Following the previous analysis for $T = 1$ and $T = 2$, and by induction on T, the problem of minimizing ϕ can be written as a dynamic program. The optimum is given by $\phi^* = \varphi_1^*$, with

$$\begin{aligned}\varphi_k^* &= \inf_{u_k} \left(\mathbf{E}((y_k - y_k^{\mathrm{des}})^2) + \mathbf{E}\left(\varphi_{k+1}^*\right)\right) \\ &= \inf_{u_k}\left(\underbrace{u_k^T \Pi_{k-1}^{-1} u_k}_{1} + \underbrace{(\hat{b}_{k-1}^T u_k - y_k^{\mathrm{des}})^2}_{2} + \underbrace{\sigma_e^2}_{3} + \mathbf{E}(\varphi_{k+1}^*)\right),\end{aligned}$$

for $k = 1, \ldots, T$, and $\varphi_{T+1}^* = 0$. All infimums are over the space of feasible policies, *i.e.*, over all u_k measurable $\sigma(y_1, \ldots, y_{k-1})$. The three marked terms can be interpreted as 1) the cost due to inaccuracy in the estimate of b, 2) the cost due to the perturbation introduced to improve estimation of b, and 3) the cost due to output noise. To make the dynamic program tractable we take the same approach as before, and use the approximation

$$\hat{b}_1 \approx \hat{b}_0, \quad \hat{b}_2 \approx \hat{b}_0, \quad \cdots \quad \hat{b}_{T-1} \approx \hat{b}_0.$$

With this approximation, we can remove the nested conditional expectation, and group the inf operators, so that

$$\phi^* \approx \inf_{u_1, \ldots, u_T} f_T,$$

with

$$f_T \triangleq \sum_{k=1}^T u_k^T \Pi_{k-1}^{-1} u_k + \sum_{k=1}^T (\hat{b}_0^T u_k - y_k^{\mathrm{des}})^2 + T\sigma_e^2.$$

Finding this minimum is not a convex program, which greatly limits our ability to solve large scale problems in practice.

5.5 Convex approximation (linearization of Π_k)

A convex approximation of the objective function above can be obtained by linearizing the information matrix in the inputs. Writing $U_k = U_k^0 + \Delta U_k$, and for $\|\Delta U_k\|$ small,

$$\begin{aligned}U_k^T U_k &\approx (U_k^0)^T U_k^0 + (U_k^0)^T \Delta U_k + \Delta U_k^T U_k^0 \\ &= (U_k^0)^T U_k + U_k^T U_k^0 - (U_k^0)^T U_k^0.\end{aligned}$$

Likewise,

$$\Pi_k \approx \Pi_0 + \sigma_e^{-2}\left((U_k^0)^T U_k + U_k^T U_k^0 - (U_k^0)^T U_k^0\right) \triangleq P_k.$$

The term omitted is $O(\sigma_e^2 \|\Delta U_k\|^2)$. It is positive semidefinite, hence the approximation undervalues information. We can expect that a solution based on this approximation will be conservative in the introduction of perturbations for the purpose of identification.

The problem now involves a sum of matrix fractional and quadratic terms, all of which are convex,

$$\text{minimize} \quad \sum_{k=1}^T u_k^T P_{k-1}^{-1} u_k + \sum_{k=1}^T (\hat{b}_0^T u_k - y_k^{\mathrm{des}})^2$$

where P_{k-1} is as above, and the variables are $u_1, \ldots, u_T \in \mathbf{R}^n$. This is a matrix-fractional and second-order cone program, which is equivalent to the semidefinite program

$$\begin{aligned}&\text{minimize} \quad \sum_{k=1}^T (\alpha_k + \beta_k) \\ &\text{subject to} \\ &\begin{bmatrix} \alpha_k & (\hat{b}_0^T u_k - y_k^{\mathrm{des}}) \\ (\hat{b}_0^T u_k - y_k^{\mathrm{des}}) & 1 \end{bmatrix} \succeq 0, \quad k = 1, \ldots, T \\ &\begin{bmatrix} \beta_k & u_k^T \\ u_k & P_{k-1} \end{bmatrix} \succeq 0, \quad k = 1, \ldots, T \\ &P_{k-1} = \Pi_0 + \sigma_e^{-2} \sum_{j=1}^{k-1} \left(u_j (u_j^0)^T + u_j^0 u_j^T - u_j^0 (u_j^0)^T\right), \\ &\qquad k = 1, \ldots, T,\end{aligned}$$

where the variables are $\alpha_1, \ldots, \alpha_T, \beta_1, \ldots, \beta_T \in \mathbf{R}$, and $u_1, \ldots, u_T \in \mathbf{R}^n$. Algorithms for solving semidefinite programs are of polynomial complexity. The complexity of solving this particular problem with an interior-point method is bounded by $O\left(T^{\frac{7}{2}} n^{\frac{9}{2}}\right)$. For more on semidefinite programming see, *e.g.*, Vandenberghe and Boyd [14].

5.6 Algorithm

A possible practical algorithm is as follows.

1. Find a nominal input sequence $u_1^0, \ldots, u_T^0$ according to a simple policy, such as minimizing $\sum_{k=1}^T u_k^T \Pi_0^{-1} u_k + \sum_{k=1}^T (\hat{b}_0^T u_k - y_k^{\mathrm{des}})^2$. This amounts to solving without accounting for the benefits of extra information.

2. Linearize the information matrices $\Pi_1, \ldots, \Pi_{T-1}$ around the nominal input sequence, to obtain the affine functions $P_1(u_1), \ldots, P_{T-1}(u_1, \ldots, u_{T-1})$. (To avoid the obvious convergence problems that occur when $u_k^0 = 0$, we add a small random term to the nominal input sequence before linearizing.)
3. Solve the semidefinite program above, to obtain a new nominal input sequence.
4. Relinearize around the new nominal input sequence and repeat the optimization. This may be repeated for a fixed number of times or until convergence (numerical experiments have shown convergence after a very small number of iterations).
5. Apply the first input of the resulting input sequence to the system, measure the output, update the distribution of b, and repeat with horizon $T \leftarrow T-1$.

(For the receeding horizon case, instead of repeating with a decreasing horizon, a new desired output y_{T+1}^{des} is introduced after application of u_1.)

Example
As a numerical example, consider the problem described for the dithering example in §4.1, with the same simulation methodology. We saw then that the expected cost with regularized passive learning was 9.3 ($\pm$0.43), and that the expected cost with the best dithering level was 4.0 ($\pm$0.19). With the algorithm described here, the expected cost is 2.0 ($\pm$0.08).

6 Conclusions

While the computation of the exact solution to the simultaneous estimation and optimization problem seems to be fundamentally intractable, the mathematical tools and computing resources now available should allow us to solve effective approximations of the problem in real-time for many applications. In this paper, we have described some early results for a simple class of problems. This class is nevertheless complex enough to explore the key ideas involved, and straightforward extensions include the class of finite impulse response dynamic systems. Numerical examples have shown that, at least in some cases, the approximation introduced can perform vastly better than standard adaptive control techniques. Future research will look into extending these results, in particular for wider classes of problems, and into developing a better understanding of the properties of the different heuristics and approximations.

References

[1] Brian D. O. Anderson. Adaptive systems, lack of persistency of excitation and bursting phenomena. *Automatica*, 21(3):247–258, 1985.

[2] A. A. Fel'dbaum. Theory of dual control, I. *Automat. Remote Control*, 21(9):1240–1249, 1960.

[3] A. A. Fel'dbaum. *Optimal Control Systems*. Academic Press, New York, 1965.

[4] Y. Bar-Shalom. Stochastic dynamic programming: Caution and probing. *IEEE Trans. Aut. Control*, AC-26(5):1184–1195, 1981.

[5] P. Dersin, M. Athans, and D. Kendrick. Some properties of the dual adaptive stochastic control algorithm. *IEEE Trans. Aut. Control*, AC-26(5):1001–1008, 1981.

[6] P. Mookerjee and Y. Bar-Shalom. An adaptive dual controller for a MIMO-ARMA system. *IEEE Trans. Aut. Control*, 34(7):795–800, 1989.

[7] P. R. Kumar and P. Varaiya. *Stochastic Systems, Estimation, Identification and Adaptive Control*. Prentice-Hall, New Jersey, 1986.

[8] D. Bertsekas. *Dynamic Programming and Optimal Control*. Athena Scientific, Belmont, Massachusetts, 1995.

[9] H. Genceli and M. Nikolaou. New approach to constrained predictive control with simultaneous model identification. *AIChE Journal*, 42(10):2857–2868, October 1996.

[10] B. L. Cooley and J. H. Lee. Control-relevant experiment design for multivariable systems. Draft, contact: jhl@eng.auburn.edu, 1997.

[11] V. Wertz R. R. Bitmead, M. Gevers. *Adaptive Optimal Control : The Thinking Man's GPC*. Prentice Hall, New Jersey, 1990.

[12] L. Ljung. *System Identification: Theory for the User*. Prentice-Hall, 1987.

[13] S. Boyd, C. Crusius, and A. Hansson. Control applications of nonlinear convex programming. *Journal of Process Control*, 8(5-6):313–324, 1998. Special issue for papers presented at the 1997 IFAC Conference on Advanced Process Control, June 1997, Banff.

[14] L. Vandenberghe and S. Boyd. Semidefinite programming. In *Siam Review*, 1995.