Published in final edited form as:
Int J Adapt Control Signal Process. 2010 March 1; 24(3): 155–177. doi:10.1002/acs.1094.

IMPLICIT DUAL CONTROL BASED ON PARTICLE FILTERING AND FORWARD DYNAMIC PROGRAMMING

David S. Bayard[1,3] and **Alan Schumitzky**[1,2]

[1] Laboratory of Applied Pharmacokinetics, University of Southern California School of Medicine, 2250 Alcazar St. CSC 134-B, Los Angeles, CA 90033

[2] University of Southern California, Mathematics Department, Los Angeles, CA 90089

Abstract

This paper develops a sampling-based approach to implicit dual control. Implicit dual control methods synthesize stochastic control policies by systematically approximating the stochastic dynamic programming equations of Bellman, in contrast to explicit dual control methods that artificially induce probing into the control law by modifying the cost function to include a term that rewards learning. The proposed implicit dual control approach is novel in that it combines a particle filter with a policy-iteration method for forward dynamic programming. The integration of the two methods provides a complete sampling-based approach to the problem. Implementation of the approach is simplified by making use of a specific architecture denoted as an H-block. Practical suggestions are given for reducing computational loads within the H-block for real-time applications. As an example, the method is applied to the control of a stochastic pendulum model having unknown mass, length, initial position and velocity, and unknown sign of its dc gain. Simulation results indicate that active controllers based on the described method can systematically improve closed-loop performance with respect to other more common stochastic control approaches.

Keywords

implicit dual control; particle filtering; policy iteration; stochastic optimal control; dynamic programming

1 INTRODUCTION

Recent literature has seen the emergence of sampling methods capable of approximating solutions to a wide range of problems previously considered intractable [27][36]. Sampling methods will continue to become more attractive with the availability of increasingly more powerful computing hardware. In light of these new developments, it is potentially beneficial to revisit old and challenging problems from the control literature.

In this paper, a sampling method is introduced for approximating the closed-loop solution to the nonlinear stochastic control problem. The proposed method can be considered as a form of implicit dual control since it acts systematically to approximate the Stochastic Dynamic Programming (SDP) equations of Bellman. This is in contrast to explicit dual control methods that induce probing into the control law by artificially changing the cost function.

[3]Correspondence to: David S. Bayard, MS 198-326, Jet Propulsion Laboratory, 4800 Oak Grove Drive, CA 91109, USA; david.bayard@jpl.nasa.gov; Phone 818-354-8208; FAX 818-957-2763.

NIH-PA Author Manuscript
NIH-PA Author Manuscript

In general, dual controllers have the desired property of active learning, that is, they optimally proportion their effort between controlling the plant, and actively probing the plant to extract useful information.

The proposed sampling method for dual control is based on combining particle filtering [55] with the Iteration in Policy Space (IPS) algorithm [15][12]. Particle filtering is emerging as the sampling method of choice for solving a broad class of nonlinear estimation problems. The IPS algorithm is a sampling method for forward dynamic programming that approximates the SDP solution using policy iteration. Combining these two approaches gives an overall sampling method for dual control that, in principle, can be applied to a wide range of nonlinear stochastic control problems.

In [56] particle filtering is discussed in the context of generating control policies of the feedback type (e.g., heuristic certainty equivalent, and open-loop feedback policies). The current paper extends these results by generating dual controllers that are of the closed-loop type. This extension is important because closed-loop control policies generally exhibit improved performance due to their property of active learning.

Optimal stochastic control is discussed in Section 2, and particle filtering is discussed in Section 3. Implicit dual control based on the IPS algorithm is discussed in Section 4, and is used to develop sampling-based dual control methods in Section 5. A stochastic pendulum is introduced in Section 6 as a model useful for studying both estimation and control. The pendulum model is used to study particle filtering in Section 7, and dual control in Section 8. Results are encouraging, indicating that active controllers based on sampling methods are capable of systematically improving performance relative to non-active control policies. Conclusions are postponed until Section 9. All results from this paper were first reported in a departmental report [18].

2 OPTIMAL STOCHASTIC CONTROL

2.1 Problem Statement

Consider the following discrete-time state and measurement equations,

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

$$y_k=h_k(x_k,v_k) \tag{2.2}$$

Here, $x \in R^{n_x}$ is the state, $u \in R^{n_u}$ is the control, $w \in R^{n_w}$ is the process noise, $y \in R^{n_y}$ is the measurement, and $v \in R^{n_v}$ is the measurement noise. The random quantities $\{w_i\}$, $\{v_i\}$ are assumed to be independent zero-mean white noise sequences and jointly independent of the random initial condition x_0. The noise and initial state statistics are assumed to be known and specified by the following probability densities,

$$x_0 \sim p_x(x_0), \;\; w_k \sim p_w(k,w_k), \;\; v_k \sim p_v(k,v_k) \tag{2.3}$$

It is desired to minimize the following expected cost criteria,

$$J(u)=E[L] \tag{2.4}$$

$$L=g_N(x_N)+\sum_{i=0}^{N-1} g_i(x_i, u_i, w_i) \tag{2.5}$$

over a class of admissible control policies. Here, g_i, $i = 0, \ldots, N$ are specified weighting functions. It is convenient for later use to define a truncated cost structure starting at time k,

$$L_k=g_N(x_N)+\sum_{i=k}^{N-1} g_i(x_i, u_i, w_i) \tag{2.6}$$

Let the information state I_k at time k be defined by,

$$I_k=[y_k, \ldots, y_0, u_{k-1}, \ldots, u_0] \tag{2.7}$$

$$I_0=[y_0] \tag{2.8}$$

The information state I_k summarizes all measurement information causally available at time k. An admissible policy is defined by a sequence of controls $\Pi = [u_0(I_0), \ldots, u_{N-1}(I_{N-1})]$ where each control u_k maps the information state I_k into a constrained space of allowable inputs $\mathcal{B}(I_k)$, i.e.,

$$u_k(I_k) \in \mathcal{B}(I_k) \subseteq R^{n_u} \tag{2.9}$$

2.2 Stochastic Dynamic Programming (SDP)

The admissible control policy that minimizes (2.4) is denoted as,

$$\Pi^{CLO}=[u_0^{CLO}(I_0), \ldots, u_{N-1}^{CLO}(I_{N-1})] \tag{2.10}$$

where “CLO” stands for Closed-Loop Optimal [10]. Using the principle of optimality, it can be shown that the CLO control policy satisfies the following stochastic dynamic programming equations of Bellman,

$$\begin{aligned} J_{N-1}^{CLO}(I_{N-1}) &= \min_{u_{N-1}} E[g_{N-1}+g_N|I_{N-1}] \\ &\vdots \\ J_k^{CLO}(I_k) &= \min_{u_k} E[g_k+J_{k+1}^{CLO}(I_{k+1})|I_k] \\ &\vdots \\ J_0^{CLO}(I_0) &= \min_{u_0} E[g_0+J_1^{CLO}(I_1)|I_0] \end{aligned} \tag{2.11}$$

and the total cost is given by,

$$J^{CLO}=E[J_0^{CLO}(I_0)] \tag{2.12}$$

The information state I_k in (2.7) can be written recursively in time,

$$I_{k+1}=(I_k, y_{k+1}, u_k) \tag{2.13}$$

This relation serves as an alternative state equation replacing (2.1), where I_k now plays the role of the state, and the quantity y_{k+1} plays the role of process noise (a more general state-dependent definition of process noise is used in [21] to make this interpretation precise). Since the information state I_k in (2.13) is updated using available information y_{k+1}, u_k, it is considered *fully observed*. This is in contrast to the state x_k in (2.1) that is only *partially observed* through the noisy measurement (2.2).

2.3 Stochastic Control Policies

A general overview of stochastic control policies is given in [10]. There are three main classes of stochastic control policies: the Open-Loop (OL) class, the Feedback (F) class, and the Closed-Loop (CL) class.

The Open-Loop (OL) policy uses only prior information, and computes the control without using any measurement information. Because measurements are not used, no learning takes place. The Feedback (F) policy determines the control input at each stage k using all measurements gathered up until time k (i.e., feedback from measurements), but does not anticipate that future measurements will be made. Since F policies learn from measurements, they use feedback and are generally known to perform better than OL policies. In certain cases this improvement can be proved theoretically [14][21].

Two commonly used F policies are the open-loop feedback (OLF) policy (originally denoted as OLOF in [28]), and the heuristic certainty equivalence (HCE) policy. The OLF policy at each time k is derived by solving for the OL control sequence $u_k, \ldots, u_N$ using all the information obtained up to time k as the prior, and then applying only the first control u_k to the plant. Since the OLF policy calculates a new open-loop control sequence at each time k, it makes use of both open-loop and feedback notions, and hence its name. The control policies developed in [9][61] are of the OLF type.

The HCE policy is generated by first solving the underlying deterministic optimal control problem (obtained by setting all random variables to their mean values with probability one, and assuming that the full state x is measured perfectly), to give the deterministic feedback relation $u_k = \varphi(x_k)$. The HCE policy is then defined by substituting the conditional-mean state estimate $\hat{x}_k = E[x_k|I_k]$ for the true state x_k to give the stochastic policy $u_k^{HCE}=\varphi(\widehat{x_k})$. Interestingly, the HCE policy is known to be optimal for the Linear Quadratic Gaussian (LQG) problem [21], the Linear Quadratic Gaussian-Sum (LQGS) problem [2], and for other restricted classes of problems [11]. While HCE is usually not optimal for more general systems, it is often used as a heuristic method to generate potentially useful suboptimal control policies [21]. For example, most modern indirect model reference adaptive control (MRAC) schemes [38] and self-tuning regulators (STRs) [39] are of the HCE type, since they substitute estimates for true parameters in deterministically-derived control laws.

The optimal policy that minimizes the expected performance cost, denoted earlier as the Closed-Loop Optimal (CLO) policy (2.10), is known to belong to the CL class [10]. The CL policy, like the F policy, determines the control at each stage k using all measurements gathered up until time k, but in addition, anticipates the fact that future measurements will be made. The anticipation of future measurements induces the CL policy to actively probe the system for new information. This intentional probing action, or “active learning feature”, is a key property of CL policies. Because probing action tends to “shake up” the plant, it is often in direct conflict with the immediate goals of controlling the plant. Consequently, CL policies are sometimes called “dual” controllers. This term was originally introduced by Feldbaum [29][30], who noted the dual character of the optimal policy in controlling the state, while simultaneously regulating its learning for control purposes. Surveys on dual control include the papers [32][66][67] and a recent book on the subject [33].

A dual controller might either dither the control input, or might use larger and/or more dynamic inputs to excite the system and better learn the plant dynamics, while simultaneously controlling the plant’s behavior. While F policies also learn, they only do so by making mistakes. Such learning is strictly accidental and not the result of planned probing actions. Accordingly, CL policies have the potential to improve significantly on the performance of F class policies.

In general, the computation of the optimal CL policy (i.e., the CLO policy) requires solving the stochastic dynamic programming (SDP) equations [8][10][19]. Unfortunately, any direct solution to the SDP equations requires overcoming the “curse of dimensionality” [19], and is for the most part computationally intractable. To date, numerical solutions have been computed for only the simplest of scalar systems [6][7][34][42]. The difficultly involved in solving the SDP equations has led researchers to look for simpler methods for generating dual control policies. Current practical approaches to dual control can be divided into two main categories: *implicit* dual and *explicit* dual.

Implicit dual control methods apply approximations to the SDP equations to obtain actively adaptive suboptimal control policies that have desired probing properties, and improved performance. In contrast, explicit dual approaches modify the cost function to include extra terms that reflect the information gathered from future measurements. Upon minimization of the overall cost, these extra terms artificially induce probing action into the controller. The control policies developed in [3][31][49][52] are of the explicit dual type. These and other explicit dual controllers are discussed in the survey literature [32][33][66][67]. The main focus of the current paper is on implicit dual controllers, to be discussed next.

It is known that each minimizing control u_k^{CLO} depends on I_k only through the conditional density $p(x_k|I_k)$ [21]. This fact establishes an important link between the fields of stochastic control and nonlinear estimation. The conditional density is known to propagate according to a recursive equation of the form,

$$p(x_{k+1}|I_{k+1})=\mathcal{F}\{p(x_k|I_k), y_{k+1}, u_k\} \tag{2.14}$$

Unfortunately, the mechanization of (2.14) is often intractable due to the need to calculate multidimensional integrals. However, a key problem studied in nonlinear estimation is the systematic approximation of the conditional density $p(x_k|I_k)$ for the purpose of developing practical recursive filters. Arguably, the most useful recursive filters to have emerged from decades of research on this problem are the Extended Kalman Filter (EKF) [35][55], the Gaussian Sum Filter (GSF) [4][59], the Multiple Model (MM) filter [47][51], and recently, the Particle Filter (PF) [5][27][40][55]. As discussed below, each of these filtering methods has been applied by researchers to address the stochastic control problem.

The EKF propagates two central moments (conditional mean and covariance), and has given rise to stochastic controllers derived based on a *wide-sense* (WS) approximation. Wide-sense dual controllers have been successfully developed in the literature [10][62] [63] [64]. Related approaches that modify the problem statement to make the wide-sense approximation an exact sufficient statistic are given in [45][49][54][60].

The GSF is a recursive filter [4][59] that makes a Gaussian-sum approximation to $p(x|I_k)$. Implicit dual controllers based on the GSF have been developed in [1], and shown by simulation to have improved performance compared to F-class policies.

In a multiple-model problem, the state is decomposed as $x = [\xi, \theta]$ where ξ propagates as a conditionally linear gaussian state-space system, and θ is a constant (but unknown), discrete-valued parameter vector belonging to a finite set $\theta \in \Theta = \{\theta^1, \ldots, \theta^s\}$. Implicit dual controllers based on the MM structure have been developed in [22][23][26] [48] [65], and have been shown by simulation to improve on F class policies.

The PF recursive filter is a relatively recent development that holds considerable promise for computing solutions to complex nonlinear estimation problems [27][40][55]. Consequently, the choice of the PF approximation as a sufficient statistic for solving stochastic control problems offers exciting new possibilities for controlling a wide range of nonlinear stochastic systems. To date, the application of PF to stochastic control has been limited to non-dual policies such as HCE and OLF [56]. The current paper aims to fill this gap by providing an approach to implicit dual control based on the PF approximation.

Compared to other approximations, the particle approximation has the advantages of capturing the multimodal and non-Gaussian character of the underlying conditional density, as well as being applicable to more challenging nonlinear problems that cannot be reliably linearized or approximated by an EKF. The PF approximation does not rely on linearization, and so does not break down when nonlinearities become dominant or when statistical variances become large. While in principle the GSF approximation offers similar advantages, the PF is considerably simpler to implement since it invokes simulation rather than a large bank of EKFs. Compared to GSF, the PF approximation also has the advantage of handling large variances without requiring periodic re-initialization [4]. However, the most important aspect of the PF approximation may be that it is sample-based and integrates well with other available sample-based methods for dual control synthesis [12][13][15].

3 PARTICLE FILTER

3.1 Background

Nonlinear estimation is concerned with the problem of mapping the conditional probability $p(x_k|I_k)$ at time k into the conditional probabiliy $p(x_{k+1}|I_{k+1})$at time $k + 1$, given the most recently measured quantities y_{k+1} and u_k. The nonlinear estimation process can be realized in two successive steps [55],

$$p(x_{k+1}|I_k,u_k)=\int p(x_{k+1}|x_k,u_k)p(x_k|I_k)dx_k \tag{3.1}$$

$$p(x_{k+1}|I_{k+1})=\frac{p(y_{k+1}|x_{k+1})p(x_{k+1}|I_k,u_k)}{\int p(y_{k+1}|x_{k+1})p(x_{k+1}|I_k,u_k)dx_{k+1}} \tag{3.2}$$

Equations (3.1) and (3.2) are commonly referred to as the *time update* and *measurement update*, respectively. They can be combined to give the single functional equation (2.14).

Particle filtering has been developed recently as an approach to approximating the solution to the nonlinear estimation problem. In particle filtering, at each stage k, the conditional probability $p(x_k|I_k)$ is approximated by a lumped-mass representation defined by a set of m particles in the particle set $\Omega\{x_k^j\}_{j=1}^s$, each of equal weight, $1/s$. Conceptually, these particles can be thought of s samples drawn from the density $p(x_k|I_k)$, whereby a histogram made from the samples would reveal a direct visualization of $p(x_k|I_k)$. Mathematically, the particle approximation $\Omega\{x_k^j\}_{j=1}^s$ to the density $p(x_k|I_k)$ can be written as,

$$p(x_k|I_k)\simeq\frac{1}{s}\sum_{j=1}^{s}\delta(x_k-x_k^j) \tag{3.3}$$

where the delta function notation $\delta(x - x_0)$ denotes a unit mass at location x_0.

Consistent with the functional equation (2.14) for nonlinear estimation, the particle set $\Omega\{x_k^j\}_{j=1}^s$ at time k representing $p(x_k|I_k)$ is updated using the latest information y_{k+1}, u_k to become the new particle set $\Omega\{x_{k+1}^j\}_{j=1}^s$ representing $p(x_{k+1}|I_{k+1})$. One of the simplest particle filter methods to perform this updating is the sampling importance resampling (SIR) filter [55],

$$\Omega\{x_{k+1}^j\}_{j=1}^s=\mathcal{F}\left\{\Omega\{x_k^j\}_{j=1}^s,y_{k+1},u_k\right\} \tag{3.4}$$

- FOR $j = 1 : s$
  - – Draw $x_{k+1}^j \sim p(x_{k+1}|x_k^j,u_k)$
  - – Calculate $\tilde{w}_{k+1}^j=p(y_{k+1}|x_{k+1}^j)$
- END FOR
- Calculate total weight: $t=\sum_{j=1}^{s}\tilde{w}_{k+1}^j$

- Normalize weights: $\mathrm{w}_{k+1}^{j}=\tilde{\mathrm{w}}_{k+1}^{j}/t$
- Resample

$$\Omega\{x_{k+1}^{j}\}_{j=1}^{s}=\text{RESAMPLE}\left\{\Omega\{x_{k+1}^{j},\mathrm{w}_{k+1}^{j}\}_{j=1}^{s}\right\}$$

The notation $\Omega\{x^{j},\mathrm{w}^{j}\}_{j=1}^{s}$ denotes that the *j*'th particle x^j has weight w^j. The notation $\Omega\{x^{j}\}_{j=1}^{s}$ having a single argument is a simplification that indicates all particles have equal weights, i.e., $\mathrm{w}^j = 1/s$, for all *j*. The operation called "RESAMPLE" simply draws *m* random samples from the lumped-mass distribution defined by the specified particle set. Specifically, given a particle set $\Omega\{x_{k+1}^{j},\mathrm{w}_{k+1}^{j}\}_{j=1}^{s}$, RESAMPLE maps the *s* particles with weights w^j, into *s* new particles having equal weights $\mathrm{w}^j = 1/s$, $j = 1, \ldots, s$. Many methods for resampling exist in the literature. To minimize computation in the current application, the Systematic Resampling method of Kitagawa [44] is chosen because it has complexity $\mathcal{O}(s)$.

After update, the particle set $\Omega\{x_{k+1}^{j}\}_{j=1}^{s}$ provides the lumped-mass approximation to the conditional density $p(x_{k+1}|I_{k+1})$,

$$p(x_{k+1}|I_{k+1})\simeq\frac{1}{s}\sum_{j=1}^{s}\delta(x_{k+1}-x_{k+1}^{j}) \tag{3.5}$$

This process is repeated for each *k* to propagate the conditional density.

3.2 Particle Filtering in Stochastic Control

The particle set $\Omega_k\{x_k^{j}\}_{j=1}^{s}$ serves as an approximate sufficient statistic replacing the conditional density $p(x_k|I_k)$. A stochastic control framework based on particle-filtering is shown in Figure 3.1 Here, the control input becomes a function of the current particle set,

$$u_k=u_k(\Omega_k\{x_k^{j}\}_{j=1}^{s}) \tag{3.6}$$

This restricted form of the controller reduces the dimensionality of the underlying state from I_k which is of growing dimension, or from the equivalent representation of the state as $p(x_k|I_k)$ which has infinite dimensions. The advantages of a finite dimensional approximation to the state that does not grow with time has been discussed in [62]. Specific use of the particle set to fulfill this role, has been suggested earlier in Salmond and Gordon [56] which develops HCE and OLF control policies based on the particle approximation. The current paper extends the application of particle filters to developing implicit dual controllers.

3.3 Dealing with Constant Parameters

One difficultly that arises in particle filtering is when a subset of the state vector *x* corresponds to a set of constant parameters. Let *θ* denote a vector of such parameters with the corresponding dynamics,

$$\theta_{k+1}=\theta_k \tag{3.7}$$

Having no process noise on the right hand side of (3.7) causes difficulties in particle filtering due to a phenomena called sample impoverishment [55][50]. This is an undesirable behavior where all particles collapse into a single particle. While various methods have been developed to address sample impoverishment, the problem is very challenging when process noise is completely absent, and there are few general results.

One common approach to try to "fix" (3.7) is to add a small amount of process noise,

$$\theta_{k+1}=\theta_k+w_k \tag{3.8}$$

The process noise w_k added is assumed to be white, zero-mean, and with Gaussian statistics,

$$w_k \sim N(0, W_k) \tag{3.9}$$

The presence of process noise helps avoid sample impoverishment and improves the overall robustness of the particle filter. However, the method becomes suboptimal since adding process noise introduces an artificial dilution of information over time that is not part of the original problem statement. Instead of (3.8), the current paper uses a method due to Lui and West [50] to address this problem.

The main insight of Lui and West [50] is to replace (3.8) by,

$$\theta_{k+1}=a\theta_k+(1-a)\overline{\theta}_k+w_k \tag{3.10}$$

where,

$$\overline{\theta}_k=E[\theta_k|I_k] \tag{3.11}$$

$$w_k \sim N(0,(1-a^2)V_k) \tag{3.12}$$

$$V_k \triangleq E[(\theta_k-\overline{\theta}_k)^2|I_k] \tag{3.13}$$

Here, $\bar{\theta}_k$ and V_k are computed from the corresponding particle averages at time k. In this case, process noise has been added on the right hand side, but the shrinkage of the particles towards the ensemble mean re-establishes invariance of the first two moments. Specifically, for any choice of $0 < a \leq 1$, it can be verified that the choice of process noise variance $Cov[w_k] = (1 - a^2)V_k$ ensures that, $E[\theta_{k+1}|I_k] = E[\theta_k|I_k]$ and $Var[\theta_{k+1}|I_k] = Var[\theta_k|I_k]$. If the statistics of θ_k were Gaussian, there would be no loss of information in the resulting particle representation of θ_{k+1}. However, in the more typical situation where the statistics of θ_k are non-Gaussian, only the first two moments remain unaffected and higher-order moments degrade accordingly. The method of Liu and West is used in the current paper to deal with the issue of constant parameters. It has been found to work well within the boundaries of the current studies.

A question that arises in practice is how to choose the shrinkage parameter a in (3.10). Guidelines are given in [50]. However, our experience indicates that the parameter a is best found by simulation experiments. A simple approach is systematically to decrease the shrinkage parameter a from unity until particle impoverishment is no longer observed in representative simulations. The value of a is then not increased past this point since the propagated distribution would degrade unnecessarily.

To model positive-valued physical parameters $p > 0$, the current paper will make use of log-Normal variates of the form $p = e^{\theta}$ where $\theta \sim N(\mu, \sigma^2)$. Consider the constant dynamics $p_{k+1} = p_k$. To modify the dynamics of a log-Normal variate p_k in the Liu-West sense, it is best to modify its Normal part as,

$$\theta_k = log(p_k) \tag{3.14}$$

$$\theta_{k+1} = a\theta_k + (1-a)\bar{\theta}_k + w_k \tag{3.15}$$

$$p_{k+1} = e^{\theta_{k+1}} \tag{3.16}$$

As desired, this approach ensures that the propagated variate p_{k+1} remains positive-valued. In addition, this approach extends the Lui-West zero-information loss property for Normal variates to include log-Normal variates.

4 IPS ALGORITHM

The IPS algorithm is a method for on-line implicit dual control that achieves its performance advantages by successively improving on a given policy [15][12]. The improvement is due to a policy iteration approach defined by determining the present control that optimizes the cost at the current stage plus the future cost-to-go as evaluated on a specified nominal policy. In this manner, the future is seen through the costs incurred by the nominal policy. The notion of policy iteration is made precise by the following definition.

DEFINITION 4.1

A policy $\Pi^{*p+1} = [u_0^{*p+1}(I_0), \ldots, u_{N-1}^{*p+1}(I_{N-1})]$ is said to be a policy iteration with respect to policy iteration with respect to policy $\Pi^{*p} = [u_0^{*p}(I_0), \ldots, u_{N-1}^{*p}(I_{N-1})]$ if at every k and I_k they are related as,

$$u_k^{*p+1}(I_k) = \begin{cases} \min_{u_k} E\left[g_k(x_k, u_k, w_k) + g_N(x_N) + \sum_{i=k+1}^{N-1} g_i(x_i, u_i^{*p}(I_k), w_i) | I_k \right] & \text{for } k=0,\ldots,N-2 \\ \min_{u_{N-1}} E\left[g_N(x_N) + g_{N-1}(x_{N-1}, u_{N-1}(I_{N-1}), w_{N-1}) | I_{N-1} \right] & \text{for } k=N-1 \end{cases} \tag{4.1}$$

The policy iteration formula (4.1) is conveniently implemented using the H-block computational architecture shown in Figure 4.1. The H block is named after its resemblence to the letter "H" created by its two connections at the top and bottom. The policy Π^{*p} is supplied from the bottom, and the policy Π^{*p+1} is computed internally and output at the top. Specifically, the information state I_k is read in at the top left, and the control u_k is read out at the top right. The information states corresponding to future simulated trajectories are read out at the bottom left, and the correspoinding future controls are read in from the bottom right. The future simulated trajectories are generated as part of the computation performed inside the H block which uses Monte Carlo simulation combined with control search to find the current u_k from condition (4.1). A specific example showing the inside workings of an H-block is given in Section 5.

In general, the policy generated by a policy iteration sees a performance improvement with respect to the policy that generated it. This result is made more precise in the following result, taken from [12] without proof.

THEOREM 4.1—Assume that policy Π^{*p+1} is defined by an iteration in policy space with respect to control policy Π^{*p}. Then the following inequality holds for all I_k, $k = 0, \ldots, N - 1$,

$$J_k^{*p+1}(I_k) \le J_k^{*p}(I_k) \tag{4.2}$$

Simply stated, this means that the policy coming out the top of an H-block performs better than the policy supplied from below.

H-blocks can be vertically cascaded successively to generate a family of new policies with monotonically improving performance. This notion is summarized in the following result taken from [15] without proof.

THEOREM 4.2—Given any admissible starting policy Π^{*0} let the sequence of control policies $\Pi^{*0}, \Pi^{*1}, \ldots, \Pi^{*N}$ be defined by successive iterations in policy space. Let the total expected cost associated with each policy Π^{*p} be defined as J^{*p}, p = 0, …, *N*. *Then,*

$$J^{CLO}=J^{*N} \le J^{*N-1} \le \ldots \le J^{*1} \le J^{*0} \tag{4.3}$$

Here, J^{*0} denotes the cost of using the nominal policy Π^{*0} by itself. It is worth noting that the N'th policy iterate achieves the Closed-Loop Optimal cost regardless of the choice of the initial policy Π^{*0}.

The multiple policy iterates of Theorem 4.2 can be implemented by vertically cascading H-blocks. This implementation is shown in Figure 4.2. Except for the bottom-most H-block H^0, all H-blocks are identical and can be implemented by exactly replicating the software. The H^0 block is special in that it only outputs a nominal policy of the designer's choosing. At any given time k the CLO Policy is achieved by cascading $N - k$ H-Blocks.

When computing the p-IPS policy via an H-block cascade, the maximum number of calls is made to the last H-block when calculating the first control. This number is given approximately as [15],

$$\beta(p,0)=\frac{(2eM)^p}{\sqrt{2\pi p}}\left(\frac{N-1}{p}\right)^p \tag{4.4}$$

where M is the number of Monte Carlo trajectories used for control search in a single H-block. The larger the number of policy iterates p, the more computation. This indicates that the p-IPS policy for $p = 1, \ldots, N$ trades-off the amount of computation with degree of optimality obtained.

The IPS algorithm can be applied to completely deterministic problems by considering the special case where all random variables attain their mean values with probability one. This gives rise to a novel method for deterministic optimization that has been developed and demonstrated in [16]. In practice, the implementation of a policy iteration can be computationally intensive. To date, real-time computational constraints have limited implementions to only a single policy iteration for stochastic problems [15] and two policy iterations for determinstic problems [16]. This situation is expected to improve as computers get faster and more capable in the future.

5 DUAL CONTROL MECHANIZATION

5.1 H-Block Architecture

The policy iteration formula (4.1) is implemented using a computational structure denoted as an H-block. The H-block structure is useful because it evaluates the expectation in (4.1) using Monte Carlo simulation. The H-block is designed to receive Π^{*p} controls from the bottom, and pass out Π^{*p+1} controls from the top. For visualization, an H-block is depicted in Figure 5.1 for a relay problem having two possible control values $u = +1, -1$. The inside of the H-block provides the necessary computations to determine policy Π^{*p+1} from Π^{*p} using policy iteration. The H-block structure of Figure 5.1 modifies an earlier H-block [15] to accommodate particle filtering and Monte Carlo control search.

The computations inside an H-block are described as follows. First, the information state I_k is passed into the H-block through the top left connection. The equivalent representation of I_k is the particle set $\Omega_k\{x_k^j\}$ that approximates the conditional density $p(x_k|I_k)$. This particle set Ω_k is duplicated inside the H-block to define the set Ω_k^A used for nonlinear filtering, and the set Ω_k^B used for Monte Carlo simulation of future trajectories needed for evaluating the expected cost-to-go. The simulations begin by drawing a particle x_k^j from Ω_k^B to initialize the current state, and by setting $\Omega_k^A \leftarrow \Omega_k$ to initialize the particle filter for the new simulation run.

The trajectory is then propagated in closed-loop simulation by generating realizations of future process noise, future noisy measurements, and future controls (as requested from the H block below). The simulation is closed-loop in the sense that future measurements and controls are used for propagating the state trajectory x_ℓ^n, $\ell = k, \ldots, N$ and updating the particle filter set Ω_ℓ^A, $\ell = k, \ldots, N$ for nonlinear estimation along the simulated trajectory. The simulation continues until the last time $k = N$ at which time the cost over the simulated trajectory is computed. Monte Carlo simulations are then repeated over M particles x_k^j drawn from the particle set Ω_k^B, and for each of the two possible controls $u_k = 1$ and $u_k = -1$. When these $2M$ trajectories are completed, the cost-to-go is computed for each value of u_k and the control providing the smaller expected cost is reported out the top right of the H-block.

The H-block in Figure 5.1 applies to relay control, but is straightforward to extend to an arbitrary (finite) number of control values by generalizing the Monte Carlo search to handle multiple alternatives [37] [53].

5.2 Computational Considerations

The H-block implementation of Figure 5.1 will best approximate a theoretical policy iteration in the limit as M, and the number of partices s, become large. Systematic methods for choosing s and M remain as a subject for future investigation. However, some guidelines are provided based on experimental experience to date.

The value for s is best determined by testing the particle filter in separate simulations that evaluate estimation performance in isolation. Once s is established in this manner, its full value should be used in the H-block implementation. Attempts to lower s beyond this value have typically been met with significantly degraded stochastic control performance.

A natural upper bound for M is to choose it equal to the number of particles, i.e $M = s$, since the Monte Carlo simulations are based on the current particle set Ω_k. However, such a choice has been found by experiment usually to be excessive. In appears that large reductions in computation can be made by reducing M to a small fraction of s. The choices made here for all simulations are $s = 5000$ and $M = 5000/25 = 200$, which represents a factor of 25 reduction. Since computational time is proportional to M, this represents a factor of 25 speed-up. Further improvements are possible by using intelligent logic to stop the random search early when further iterations are not warranted. Details of the stopping logic are given in the next subsection. It has been found that the stopping logic terminates the search after an average of 82 simulations, relative to a maximum value of $M = 200$, which represents another factor of 2 speed-up. This gives an overall factor of approximately $2 \cdot 25 = 50$ in total speed-up.

The price for this speed-up is that the effectiveness of the policy iteration is reduced. However, simulation results in Section 8 show that even with this level of approximation, performance improvements can be maintained relative to nominal control policies.

5.3 Control Search Stopping Rule

For the purpose of improving computational efficiency, a special stopping rule is introduced into the control search. In the H-block of Figure 5.1, the determination of control u_k at each time k requires a search to minimize the expected cost,

$$\min_{i=1,2} E[L_k(i)] \tag{5.1}$$

where $i = 1$ corresponds to the choice $u_k = 1$, and $i = 2$ corresponds to the choice $u_k = -1$.

The two expectations in (5.1) are not known exactly, but are each approximated in the H-block using a Monte Carlo simulation over M trajectories,

$$E[L_k(i)] \simeq \widehat{J}_k(i) = \frac{1}{M}\sum_{n=1}^{M} L_k^n(i), \quad i=1,2 \tag{5.2}$$

The H-block of Figure 5.1 fixes the value of *M*. However, if the stopping rule is used, the value of *n* is increased until some point $n = m$ when a stopping rule is satisfied, or when $n = M$ is reached, whichever comes first. The stopping rule is,

$$|\widehat{d}(m)|+\delta J \geq \alpha\,\widehat{\sigma}_d(m) \tag{5.3}$$

where,

$$\widehat{d}(m) \triangleq \widehat{J}_k(2) - \widehat{J}_k(1) \tag{5.4}$$

$$\widehat{\sigma}_d^2(m)=\frac{1}{m(m-1)}\sum_{n=1}^{m}\left(L_k^n(2) - L_k^n(1) - \widehat{d}(m)\right)^2 \tag{5.5}$$

Here, $\delta J \geq 0$ and α are parameters chosen by the user. If (5.3) is satisfied, the search is stopped, and the stopping rule indicates that *a sufficiently large value of m has been obtained to ensure that the probability of making an error of more than* δJ *units of expected cost, has a probability less than* γ. For example, the choice $\alpha = 2$ gives a confidence of $\gamma = 0.0227$. The values of $\delta J \geq 0$ and α are specified by the user.

The desired properties of the stopping rule (5.3) are proved in Appendix C. The proof assumes normality of the MC estimates, so the rule should not be invoked until *m* is already sufficiently large to justify using asymptotic statistics (a value of $\underline{m} = 40$ is used in the software). Recursive expressions for $\hat{d}(m)$ and $\hat{\sigma}_d(m)$ are utilized to simplify the implementation.

The usefulness of the stopping rule (5.3) is briefly explained. Intuitively, the quantity $\hat{\sigma}_d(m)$ in (5.5) decreases asymptotically with *m*, and at some point satisfies the stopping rule (5.3). Consider the two extreme cases where $|\hat{\delta}(m)| \gg \delta J$ (Case I) and $|\hat{\delta}(m)| \ll \delta J$ (Case II). In Case I, δJ can be neglected so that the stopping rule (5.3) is approximated as,

$$|\widehat{d}(m)| \geq \alpha\widehat{\sigma}_d(m) \tag{5.6}$$

When (5.6) is satisfied, the situation looks like Figure 5.2. The peaks are sufficiently separated relative to the uncertainty to confidently decide a winner, and the search can stop.

In Case II, $\hat{\delta}(m)$ can be neglected so that the stopping rule (5.3) is approximated as,

$$\delta J \geq \alpha\widehat{\sigma}_d(m) \tag{5.7}$$

When (5.7) is satisfied, the situation looks like Figure 5.3. The peaks are closely spaced relative to the allowable error δJ, indicating that the controls essentially perform the same, and distinguishing further between their performance is not worth the effort. Cases lying between Cases I and II, benefit from both of these interpretations but in a more complex fashion.

NIH-PA Author Manuscript

NIH-PA Author Manuscript

In addition to stopping rule (5.3), a strict upper bound M is enforced on n. This means that simulations are stopped when $n = M$ regardless of whether or not condition (5.3) is satisfied.

6 PENDULUM MODEL

6.1 Physical Model

A pendulum is studied because it represents one of the most basic physical systems. A pendulum is shown in Figure 6.1. The pendulum has unknown length l, unknown mass $\mathbf{m}$, and unknown force influence coefficient b. The pendulum is assumed to obey the linear differential equations [41],

$$\mathrm{m}\ddot{\rho}+\frac{\mathrm{m}g}{l}\rho=bu \tag{6.1}$$

where g is the acceleration of gravity, and ρ denotes the horizontal displacement. Dividing both sides by $\mathbf{m}$ gives,

$$\ddot{\rho}+\frac{g}{l}\rho=\frac{b}{\mathrm{m}}u \tag{6.2}$$

The quantity b is assumed to have an unknown sign in the sense that it equals +1 or −1 with equal probability.

Define the quantities

$$\omega \triangleq \sqrt{g/l} \tag{6.3}$$

$$\beta \triangleq \frac{b}{\mathrm{m}} \tag{6.4}$$

The quantity ω is denoted as the *pendulum frequency*, while β is denoted as the pendulum *input coefficient*. It is seen that ω is a function of the pendulum's length l, while β is a function of its mass $\mathbf{m}$ and high-frequency gain b. Substituting (6.3) and (6.4) into (6.2) gives,

$$\ddot{\rho}+\omega^2\rho=\beta u \tag{6.5}$$

The distribution for ω is chosen as log-Normal with mean $\bar{\omega}$ and variance Σ_ω,

$$\omega \sim LGN(\overline{\omega}, \sum\nolimits_{\omega}) \tag{6.6}$$

The log-Normal variate ω can be formed as [25],

$$\omega = e^z \tag{6.7}$$

where $z \sim N(\mu, s^2)$ and,

$$\sigma^2 = \log(1 + \frac{\sum_\omega}{\overline{\omega}^2}) \tag{6.8}$$

$$\mu = \log(\overline{\omega}) - \frac{1}{2}\sigma^2 \tag{6.9}$$

The use of log-Normal rather than Normal ensures that the variable ω stays non-negative which is desired to ensure a physically meaningful oscillation frequency. The distribution for the input coefficient β is chosen as the two-component Gaussian mixture,

$$\beta \sim .5N(\overline{\beta}, \sum_\beta) + .5N(-\overline{\beta}, \sum_\beta) \tag{6.10}$$

This choice is consistent with the definition of $\beta = b/\mathbf{m}$ in (6.4), where a Gaussian distribution $N(\overline{\beta}, \Sigma_\beta)$ is assumed for the reciprocal mass $\mathbf{m}^{-1}$, and a Bernoulli distribution is assumed on the force influence coeffient b.

Letting $v = \dot{\rho}$, the dynamics of the pendulum can be put into state-space form as,

$$\dot{\omega} = 0 \tag{6.11}$$

$$\dot{\beta} = 0 \tag{6.12}$$

$$\begin{bmatrix} \dot{\rho} \\ \dot{v} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -\omega^2 & 0 \end{bmatrix} \begin{bmatrix} \rho \\ v \end{bmatrix} + \begin{bmatrix} 0 \\ \beta \end{bmatrix} u \tag{6.13}$$

$$y = [1, 0] \begin{bmatrix} \rho \\ v \end{bmatrix} \tag{6.14}$$

Vectorizing the physical position and velocity states as,

$$\xi = \begin{bmatrix} \rho \\ v \end{bmatrix} \tag{6.15}$$

equations (6.13) and (6.14) are conveniently expressed in matrix form as,

$$\dot{\xi} = \tilde{A}\,\xi + \tilde{B}\,u \tag{6.16}$$

$$y = \tilde{C}\,\xi \tag{6.17}$$

where,

$$\tilde{A} = \begin{bmatrix} 0 & 1 \\ -\omega^2 & 0 \end{bmatrix} \tag{6.18}$$

$$\tilde{B} = \begin{bmatrix} 0 \\ \beta \end{bmatrix}; \quad \tilde{C} = [1, 0] \tag{6.19}$$

6.2 Discretization

Assuming piecewise constant controls, the deterministic system (6.18),(6.19) can be exactly discretized with a sampling period of T seconds as,

$$\xi_{k+1} = A\xi_k + Bu_k \tag{6.20}$$

$$y_k = C\xi_k \tag{6.21}$$

where,

$$A = e^{\tilde{A}T} = \begin{bmatrix} \cos\omega T & \frac{\sin\omega T}{\omega} \\ -\omega\sin\omega T & \cos\omega T \end{bmatrix} \tag{6.22}$$

$$B = \left(\int_o^T e^{\tilde{A}\tau} d\tau\right) \tilde{B} = \begin{bmatrix} \beta\frac{(1-\cos\omega T)}{\omega^2} \\ \beta\frac{\sin\omega T}{\omega} \end{bmatrix}; \quad C = \tilde{C} = [1, 0] \tag{6.23}$$

Equations (6.20) and (6.21) will form the starting point for a stochastic control model.

6.3 Stochastic Control Model

A stochastic control model is defined by adding white process noise w_k and white measurement noise v_k to the discretized model (6.20),(6.21), and then augmenting the state vector with the constant parameters (6.11)(6.12) to give,

$$\omega_{k+1}=\omega_k \tag{6.24}$$

$$\beta_{k+1}=\beta_k \tag{6.25}$$

$$\xi_{k+1}=A_k\xi_k+B_ku_k+\Gamma_kw_k \tag{6.26}$$

$$y_k=C\xi_k+v_k \tag{6.27}$$

$$A_k=\begin{bmatrix} \cos\omega_kT & \frac{\sin\omega_kT}{\omega_k} \\ -\omega_k\sin\omega_kT & \cos\omega_kT \end{bmatrix} \tag{6.28}$$

$$\Gamma_k=B_k=\begin{bmatrix} \beta_k\frac{(1-\cos\omega_kT)}{\omega_k^2} \\ \beta_k\frac{\sin\omega_kT}{\omega_k} \end{bmatrix} \tag{6.29}$$

$$w_k \sim N(0,W);\ v_k \sim N(0,V) \tag{6.30}$$

The prior on the initial state x_0 is specified as,

$$x_0=[\omega_0,\beta_0,\rho_0,\nu_0]^T \tag{6.31}$$

$$\omega_0 \sim LGN(\overline{\omega},\sum\nolimits_{\omega}) \tag{6.32}$$

$$\beta_0 \sim .5N(\overline{\beta},\sum\nolimits_{\beta})+.5N(-\overline{\beta},\sum\nolimits_{\beta}) \tag{6.33}$$

$$\rho_0 \sim N(\overline{\rho},\sum\nolimits_{\rho}) \tag{6.34}$$

$$\nu_0 \sim N(\overline{\nu},\sum\nolimits_{\nu}) \tag{6.35}$$

NIH-PA Author Manuscript

NIH-PA Author Manuscript

where all scalar elements of the state vector are assumed to be statistically independent. Defining the augmented state vector,

$$x_k = \begin{bmatrix} \omega_k \\ \beta_k \\ \rho_k \\ \nu_k \end{bmatrix} \tag{6.36}$$

the model (6.24)–(6.27) can be written more compactly as a special case of the desired nonlinear state-space form (2.1)(2.2), for which all of the earlier control and estimation strategies are applicable.

7 CASE STUDY: Particle Filtering

7.1 Overview

In this section a particle filter is designed to perform nonlinear estimation for the pendulum model. The noise covariances are specified according to (6.30) with the choices,

$$W = (.1)^2; \; V = 1 \tag{7.1}$$

The sampling time is T = 1 and the prior on the initial state x_0 is specified according to (6.31)-(6.35) with the choices,

$$\overline{\omega} = 2\pi(.25), \; \sum\nolimits_{\omega} = (2\pi(.05))^2 \tag{7.2}$$

$$\overline{\beta} = 12, \; \sum\nolimits_{\beta} = 4 \tag{7.3}$$

$$\overline{\rho} = 0, \; \sum\nolimits_{\rho} = 4 \tag{7.4}$$

$$\overline{\nu} = 0, \; \sum\nolimits_{\nu} = (.4)^2 \tag{7.5}$$

7.2 Simulation Results

The value of the initial true state $x_0 = [\omega_0, \beta_0, \rho_0, \nu_0]^T$ is given as,,

$$\omega_0 = \omega^* = 2 \tag{7.6}$$

$$\beta_0=\beta^{*}=9 \tag{7.7}$$

$$\rho_0=1\,(m) \tag{7.8}$$

$$v_0=-.1\,(m/s) \tag{7.9}$$

Since ω^*, β^* are the true pendulum parameter values, they have been specially notated with the superscript "*".

The starting mean values of the particle filter are given as $E[\omega] = 1.5677$ and $E[\beta] = -.014172$. The value for $E[\beta]$ should be theoretically zero, but is non-zero in practice due to the use of a finite number (i.e., $s = 5000$) of particles. The control input u_k is chosen randomly at each time instant with equally probable values of +1 or −1.

The particle filter is propagated with measurements over the 20 second horizon. The Lui-West method is used with its roughening parameter chosen as $a = .95$. Results after 20 seconds are summarized in Table 7.1. For reporting purposes, the conditional-mean of the particle filter is used as a point estimate. The conditional-mean plus and minus the 50 and 95 percentile bounds are shown superimposed on truth values for the ω and β parameters in Figure 7.1 and Figure 7.2, respectively. The final estimates are given as $E[\omega|I_{20}] = 2.0139$ and $E[\beta|I_{20}] = 9.4964$. Pendulum position is shown in Figure 7.3. Position error is plotted with 50 and 95 percentile confidence bounds in Figure 7.4. It is seen that the error lies within the predicted error bounds.

8 CASE STUDY: Dual Control

8.1 Overview

Two dual control case studies are presented in this section. The goal in the first case is to achieve a terminal position of 2 m after 6 stages, and the goal in the second case is to achieve a terminal position of 4 m after 4 stages. Both cases are challenging due to the large initial uncertainty and limited control authority. However, the second case is more challenging because a larger excursion must be achieved despite having less time to learn the parameters and to elicit the desired controlled behavior.

The noise and prior statistics are same as used earlier in the particle filtering study, and are specified as (6.30)-(6.35) with the choices (7.1)-(7.5). The controls u_k are restricted to be of the relay type, having values of either 1 or − 1. The sampling time is chosen as $T = 1$ s. A digital controller is used, where the control inputs u_k are constant over each sampling period of 1 second. The control search parameters are set at $M = 200$, $\underline{m} = 40$, $\delta J = 2$, $\alpha = 2$. The particle filter for this problem uses $s = 5000$ particles, and is identical to the one used earlier in Section 7.

Two histograms are shown to help understand the control challenge. The pendulum period is shown in Figure 8.1. From this histogram, it can be seen that about half the pendulum realizations have a period with less than a single cycle contained within the 4 second controlled time-horizon. However, most pendulum realizations have a period with at least one cycle contained within a 6 second time horizon. With less than a single cycle observed, it is more challenging to learn and control the pendulum frequency over the 4 second horizon.

Set-point goals for pendulum control are taken as 2 meters and 4 meters in the two case studies, respectively. The DC gain of the pendulum is given by β/ω^2, and has a histogram shown in Figure 8.2. If the maximum control of $u = +1$ is applied as a unit step function until a steady-state condition is reached, most simulated pendulum realizations would achieve the 2 meter excursion, while only two thirds would achieve the 4 meter excursion. Clearly, the 4 meter excursion is very challenging, particularly for a controller that will not have time to reach a steady-state condition.

8.2 Case 1: Six-Stage Horizon

For the first study, there are $N = 6$ stages in the horizon. The cost is given by the terminal expression

$$L = g_6(x_6) = (\rho_6 - 2)^2 \tag{8.1}$$

The 1-IPS policy with respect to the HCE policy is denoted as the 1-IPS(HCE) policy. In this study, the performance of the HCE policy is compared to that of the 1-IPS(HCE) policy.

The HCE policy is first used to control the pendulum model. The HCE policy for the current example is provided in Appendix A. Performance is assessed by running 10, 000 Monte Carlo simulations. The final expected cost is found to be 11.087 with a 1-sigma uncertainty of ±0.20596 in the MC estimate.

The 1-IPS(HCE) policy is implemented based on the H-block configuration of Figure 5.1, using HCE as the nominal policy. Again, the particle filter is mechanized using $s = 5000$ particles. Performance is assessed by running 1, 000 MC simulations. The final cost is 8.8968 with a 1-sigma uncertainty of ±0.43626 in the MC estimate. This represents an improvement compared to the the HCE policy. For convenience, results are summarized in Table 8.2

It is useful to expand the cost J into mean and variance components as follows,

$$\begin{aligned} J = E[(\rho_6 - 2)^2] = E\left[((m_J - 2) + (\rho_6 - m_J))^2\right] \\ = E[(m_J - 2)^2] + E[(\rho_6 - m_J)^2] \\ = (m_J - 2)^2 + \sigma_J^2 \\ = \text{controller bias} + \text{controller variance} \end{aligned} \tag{8.2}$$

where $m_J = E[\rho_6]$ and $\sigma_J^2 = E[(\rho_6 - m_J)^2]$. Equation (8.2) indicates that the cost J can be decomposed into two terms. The first term $(m_J - 2)^2$ depends on how well the controlled mean m_J matches the desired goal of 2. This term is denoted as *controller bias*. The second term σ_J^2 corresponds to the controlled dispersion of ρ_6 about its own mean m_J. This term is denoted as *controller variance* with its square-root σ_J denoted as the *controller variability*. Ideally, it is desirable for a controller to keep both the controller bias and variance terms small.

Results from Case 1 can be interpreted in light of the decomposition (8.2). Specifically, the 1-IPS(HCE) policy has essentially the same controller bias as the HCE policy ($m_J = 1.59$

compared to m_J = 1.61), but improves on the cost by reducing the controller variability from σ_J = 3.31 to σ_J = 2.96.

8.3 Case 2: Four-Stage Horizon

For Case 2, the problem is made more challenging by modifying the terminal cost (8.1) to become,

$$L=g_4(x_4)=(\rho_4-4)^2 \tag{8.3}$$

Here, the horizon has been shortened from N = 6 to N = 4 stages, and the desired excursion increased from 2 to 4 meters. This is more challenging because the pendulum behavior must be learned more quickly, and controlled to swing further in a shorter time. The noise and prior statistics (7.1)-(7.5) are left unchanged.

The HCE policy is tested first using 10, 000 Monte Carlo simulations. The final cost is 18.794 with a 1-sigma uncertainty of ±0.34604 in the MC estimate. This cost is greater than for Case 1, reflecting the more challenging control problem.

The 1-IPS(HCE) policy is tested next using 1, 000 Monte Carlo simulations. The final cost is 16.536 with a 1-sigma uncertainty of ±1.0265 in the MC estimate. This represents an improvement compared to the HCE policy.

The OLF policy for the current example is defined in Appendix B, calculated using s_{OLF} = 200 particles. The OLF policy is tested using 10, 000 Monte Carlo simulations. The final cost is 15.874 with a 1-sigma uncertainty of ±0.25783 in the MC estimate. This cost is better than even the 1-IPS(HCE) policy for this problem. This motivates developing a 1-IPS policy with respect to the OLF policy.

The 1-IPS policy with respect to the OLF policy is denoted as the 1-IPS(OLF) policy. The 1-IPS(OLF) policy is tested using 1, 000 Monte Carlo simulations. The final cost is 14.8726 with a 1-sigma uncertainty of ±1.1062 in the MC estimate. This represents an improvement compared to the OLF policy. For convenience, results are summarized in Table 8.3

As in Case 1, it is useful to expand the expected cost into mean and variance components,

$$J=E[g_4(x_4)]=(m_J-4)^2+\sigma_J^2 \tag{8.4}$$

where now,

$$m_J=E[\rho_4];\ \sigma_J^2=E[(\rho_4-m_J)^2] \tag{8.5}$$

Results from Case 2 are compared graphically in Figure 8.3 and can be interpreted in light of the decomposition (8.5). The goal of 4 is shown as the dash-dot line. For each control policy, the mean position m_J (solid) at the final time is shown along with its ±1 standard deviation σ_J (upper and lower dashed line). The improvement relative to HCE from using 1-IPS(HCE) is due primarily to a reduction in controller variability σ_J from 4.30 to 4.04. Interestingly, the OLF policy has a control bias larger than the 1-IPS(HCE), but is still able to improve on overall cost by having a reduced control variability σ_J of 3.81 compared to

4.04. The 1-IPS(OLF) improves on this by keeping the control variability essentially the same at σ_J = 3.85, but by increasing the mean value m_J from 2.84 to 3.77 which reduces controller bias by being closer to the desired goal of 4. As shown in Figure 8.3, the 1-IPS(OLF) policy attains the goal with the least bias, and is essentially tie for the smallest variability, giving it the best overall cost J.

All simulations were performed in Matlab 7.0.4 on a 3 GHz Pentium-4 PC computer (I875 chipset), with 2 GB memory, and an 8 MHz front-side bus. The average time taken to calculate a single HCE control was .05 s, compared to 20 s for 1-IPS(HCE). This implies that policy iteration with respect to HCE took 20/.05=400 times longer to calculate than a single HCE control. Similarly, the average time taken to calculate a single OLF control was .08 s, compared to 20 s for 1-IPS(OLF). This implies that policy iteration with respect to OLF took 20/.08 $\simeq$ 260 times longer to calculate than a single OLF control. As pointed out in Section 5.2, the current implementation benefits from a considerable computational reduction in going from $M = s = 5000$ to $M = 200$ (a 25 times speed-up), and in using a stopping rule with parameters $\delta J = 2$, $\alpha = 2$ (approximately factor of 2 speed-up). It is conceivable that improved performance can be achieved at the expense of longer run times by increasing M and using less conservative search parameters (i.e., smaller δJ and larger α). This remains as an area for future investigation.

9 CONCLUSIONS

A sampling-based method is introduced for developing implicit dual controllers. The approach combines particle filtering for nonlinear estimation with the IPS algorithm for approximating the SDP equations of Bellman. This provides a complete sampling approach to the problem. Simulation methods effectively handle all the underlying estimation and control calculations as part of an integrated H-block data structure. Suggestions are given for reducing the H-block computational loads in practical implementations. The method is applied to a numerical example based on a pendulum having unknown parameters, random initial conditions, and unknown sign of its dc gain. The method is shown systematically to improve on standard stochastic control policies. This improvement is due to the active learning features of the synthesized control laws, in contrast to the nominal starting policies (HCE and OLF) that are known to be passive.

Future research efforts will consider applications having more than two control input values, methods to reduce computation while retaining or even improving performance, and parallel processing architectures. As computers become faster over the next decade, it may become feasible to consider cascaded H-block architectures (multiple policy iterates) for improved performance. Long term goals are to improve current approaches to pharmacokinetic control and drug administration problems [57], that are traditionally handled using non-dual stochastic control approaches (e.g., HCE in [58], and OLF in [17][43]).

Acknowledgments

This work was supported by NIH grants GM068968 and EB005803 (Dr. Roger W. Jelliffe, PI), through the USC School of Medicine, Laboratory of Applied Pharmacokinetics.

Contract/grant sponsor: National Institute of Health; contract/grant numbers: GM068968, EB005803

References

1. Alspach DL. Dual control based on approximate a posteriori density functions. IEEE Trans Automatic Control 1972;17(5):689–693.

2. Alspach DL, Sorenson H. Stochastic optimal control for linear but non-Gaussian systems. Int J Control 1971;13(6):1169–1181.
3. Alster J, Belanger P. A technique for dual adaptive control. Automatica 1974;10:627–634.
4. Anderson, BDO.; Moore, JB. Optimal Filtering. Prentice-Hall; Englewood Cliffs, New Jersey: 1979.
5. Arulampalam MS, Maskell S, Gordon N, Clapp T. A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking. IEEE Trans Signal Processing February;2002 50(2)
6. Astrom KJ, Helmersson A. Dual control of an integrator with unknown gain. Comp & Maths with Appls 1986;12A(6):653–662.
7. Astrom KJ, Wittenmark B. Problems of identification and control. J Math Anal Appl 1971;34
8. Bar-Shalom Y. Stochastic dynamic programming: caution and probing. IEEE Trans Automatic Control 1981;26(5):1184–1195.
9. Bar-Shalom Y, Sivan R. The optimal control of discrete time systems with random parameters. IEEE Trans Automatic Control 1969;14(1):3–8.
10. Bar-Shalom, Y.; Tse, E. Concepts and methods in stochastic control. In: Leondes, CT., editor. Control and Dynamics Systems. New York: Academic; 1976. p. 99-172.
11. Bar-Shalom Y, Tse E. Dual effect, certainty equivalence, and separation in stochastic control. IEEE Trans Automatic Control 1974;19(5):494–500.
12. Bayard DS, Eslami M. Implicit dual control for general stochastic systems. Optimal Control Applications and Methods 1985;6:265–279.
13. Bayard DS, Eslami M. On the evaluation of expected performance cost for partially observed stochastic systems operating in closed-loop. Int J Control 1985;42(2):443–447.
14. Bayard DS. Proof of quasi-adaptivity for the m-Measurement class of feedback control policies. IEEE Trans Automatic Control May ;1987 32(5):447–451.
15. Bayard DS. A forward method for optimal stochastic nonlinear and adaptive control. IEEE Trans Automatic Control September;1991 36(9):1046–1053.
16. Bayard DS. Reduced complexity dynamic programming based on policy iteration. J Math Anal Appl October;1992 170(1):75–103.
17. Bayard, DS.; Jelliffe, RW.; Schumitzky, A.; Milman, MH.; Van Guilder, M. Precision drug dosage regimens using multiple model adaptive control: Theory and application to simulated Vancomycin therapy. In: Sridhar, R.; Rao, KS.; Lakshminarayanan, V., editors. Selected Topics in Mathematical Physics, Prof R Vasudevan Memorial Volume. World Scientific Publishing Co; Madras: 1995.
18. Bayard DS, Schumitzky A. Implicit dual control based on particle filtering and forward dynamic programming. USC Laboratory of Pharmacokinetics. November 19;2007 Report 2007-1.
19. Bellman, R. Adaptive Control Processes: A Guided Tour. Princeton University Press; Princeton N.J: 1961.
20. Bellman, R. Dynamic Programming. Princeton University Press; Princeton N.J: 1957.
21. Bertsekas, DP. Dynamic Programming: Deterministic and Stochastic Models. Prentice Hall; Englewood Cliffs, J.J: 1987.
22. Birmiwal K. A new adaptive LQG control algorithm. Int J of Adaptive Control and Signal Processing 1994;8:287–295.
23. Birmiwal K, Bar-Shalom Y. Dual control guidance for simultaneous identification and interception. 1984;20(6):737–749.
24. Bucy, RS.; Senne, KD. Realization of optimum discrete-time nonlinear estimators. Symposium on Nonlinear Estimation Theory and its Applications; San Diego, CA. Sept. 21–23; 1970.
25. DeGroot, MH. Probability and Statistics. 2. Addison-Wesley, Reading, Mass; 1989.
26. Deshpande JG, Upadhyay TN, Lainiotis DG. Adaptive control of linear stochastic systems. Automatica 1973;9:107–115.
27. Doucet, A.; de Freitas, N.; Gordon, N. Sequential Monte Carlo Methods in Practice. Springer-Verlag; New York: 2001.
28. Dreyfus S. Some types of optimal control of stochastic systems. SIAM J Contr 1964;2:120–134.

29. Feldbaum AA. Dual control theory I-IV. Auto and Remote Contr 1961;21:874–880. 1033–1039.1962;22:1–12. 109–121.
30. Feldbaum, AA. Optimal Control Systems. Academic Press; New York: 1965.
31. Filatov, NM.; Unbehauen, H. Improved adaptive dual version of generalized minimum variance (GMV) controller. Proc. 11th Yale Workshop on Application of Adaptive Systems Theory; Yale University; 1996. p. 137-142.
32. Filatov NM, Unbehauen H. Survey of adaptive dual control methods. IEE Proc Control Theory and Applications 2000;147(1):118–128.
33. Filatov, NM.; Unbehauen, H. Adaptive Dual Control. Springer-Verlag; New York: 2005.
34. Florentin JJ. Optimal probing adaptive control of a simple Bayesian system. J Elect and control 1962;13:165–177.
35. Gelb, A. Applied Optimal Estimation. The MIT Press; Cambridge, Massachusetts: 1984.
36. Gilks, WR.; Richardson, S.; Spiegelhalter, DJ. Markov Chain Monte Carlo in Practice. Chapman & Hall; New York: 1996.
37. Goldsman D, Kim SH, Marshall WS, Nelson BL. Ranking and selection for steady-state simulation: Procedures and prospectives. INFORMS J Computing 2002;14:2–19.
38. Goodwin, GC.; Sin, KS. Adaptive Filtering Prediction and Control. Prentice-Hall; New Jersey: 1984.
39. Astrom KJ, Wittenmark B. On self-tuning regulators. Automatica 1973;9:185–199.
40. Gordon N, Salmond D, Smith AFM. Novel approach to non-linear and non-Gaussian Bayesian state estimation. Proc Inst Elect Eng, F 1993;140:107–113.
41. Halliday, D.; Resnick, R. Physics: Parts I and II. John Wiley & Sons, Inc; New York: 1966.
42. Jacobs OLR, Langdon SM. An optimal extermal control system. Automatica 1970;6:297–301.
43. Jelliffe, R.; Bayard, D.; Schumitzky, A.; Milman, M.; Jiang, F.; Leonov, S.; Gandhi, V.; Gandhi, A.; Botnen, A. Multiple Model (MM) dosage design: Achieving target goals with maximal precision. 14th IEEE Symposium on Computer-Based Medical Systems (CMBS'01); July 26–27; 2001.
44. Kitagawa G. Monte Carlo filter and smoother for non-Gaussian non-linear state space models. Journal of Computational and Graphical Statistics 1996;5(1):1–25.
45. Kulcsar C, Pronzato L, Walter E. Dual control of linearly parameterised models via prediction of posterior densities. European J Control 1996;2:135–143.
46. Kwakernaak, H. On-line dynamic optimization of stochastic control systems. Proc. Third IFAC Congress; London, England. 1966. p. 29D.1-29D.7.
47. Lianiotis, DG. Partitioning: A unifying framework for adaptive systems, I: Estimation. Proc. IEEE; 1976. p. 1126-1142.
48. Lianiotis, DG. Partitioning: A unifying framework for adaptive systems, II: Control. Proc. IEEE; 1976. p. 1182-1179.
49. Lindoff B, Holst J, Wittenmark B. Analysis of approximations of dual control. Int J of Adaptive Control and Signal Processing 1999;13:593–620.
50. Liu, J.; West, M. Combined parameter and state estimation in simulation-nased filtering. In: Doucet, A.; de Freitas, N.; Gordon, N., editors. Sequential Monte Carlo Methods in Practice. Springer-Verlag; New York: 2001.
51. Magill D. Optimal adaptive estimation of sampled stochastic processes. IEEE Trans Automatic Control 1965;10(4):434–439.
52. Milito R, Padilla CS, Padilla RA, Cadorin D. An innovations approach to dual control. IEEE Trans Automatic Control 1982;27(1):132–137.
53. Nelson BL, Swann J, Goldsman D, Song W. Simple procedures for selecting the best simulated system when the number of alternatives is large. Operations Research 2001;49:950–963.
54. Pronzato L, Kulcsar C, Walter E. An actively adaptive control policy for linear models. IEEE Trans Automatic Control 1996;41(6):855–858.
55. Ristic, B.; Arulampalam, S.; Gordon, N. Beyond the Kalman filter: Particle Filters for Tracking Applications. Artech House; Boston: 2004.

56. Salmond, D.; Gordon, N. Particles and mixtures for tracking and guidance. In: Doucet, A.; de Freitas, N.; Gordon, N., editors. Sequential Monte Carlo Methods in Practice. Springer-Verlag; New York: 2001.
57. Schumitzky, A. Stochastic control of pharmacokinetics. In: Maronde, RF., editor. Topics in Clinical Pharmacology. Springer-Verlag; New York: 1986.
58. Sheiner LB, Halkin H, Peck CP, Rosenberg B, Melmon KL. Improved computer-assisted Digoxin therapy. Ann Int Med 1975;82:619–627. [PubMed: 1137256]
59. Sorenson HW, Alspach DL. Recursive Bayesian estimation using Gaussian sums. Automatica 1971;7(4):465–479.
60. Thompson AM, Cluett WR. Stochastic iterative dynamic programming: A Monte Carlo approach to dual control. Automatica 2005;41:767–778.
61. Tse E, Athans M. Adaptive stochastic control for a class of linear systems. IEEE Trans Automatic Control 1972;17(1):38–52.
62. Tse E, Bar-Shalom Y, Meier L. Wide-sense adaptive dual control for nonlinear stochastic systems. IEEE Trans Automatic Control April;1973 18(2):98–108.
63. Tse E, Bar-Shalom Y. An actively adaptive control for linear systems with random parameters via the dual control approach. IEEE Trans Automatic Control April;1973 18(2):109–117.
64. Tse E, Bar-Shalom Y. Actively adaptive control for nonlinear stochastic systems. Proc IEEE August;1976 64(8):1172–1181.
65. Wenk CJ, Bar-Shalom Y. A multiple model adaptive dual control algorithm for stochastic systems with unknown parameters. IEEE Trans Automatic Control 1980;25(4):703–710.
66. Wittenmark B. Stochastic adaptive control methods: a survey. Int J Contr 1975;21(5):705–730.
67. Wittenmark, B. Adaptive dual control methods: an overview. 5th IFAC Symp. on Adaptive Systems in Control and Signal Processing; Budapest. 1995. p. 67-73.

A APPENDIX: HCE Control

Given the current mean state $\hat{x}_k$, the HCE control at time k for a terminal cost problem is calculated by assuming all random variables attain their mean values, and minimizing the cost,

$$\min_{U_k} g_N(\widehat{x}_N) \tag{A.1}$$

$$\widehat{x}_N = E[x_N | x_k = \widehat{x}_k, \{w_i = 0, v_i = 0, i = k, \ldots, N-1\}] \tag{A.2}$$

where the controls being optimized over are given by the open-loop sequence,

$$U_k = [u_k, u_{k+1}, \ldots, u_{N-1}]^T \tag{A.3}$$

$$u_n = \{+1, -1\}, \quad n = k, \ldots, N-1 \tag{A.4}$$

It can be shown that the terminal cost can be written in matrix form as,

$$g_N(\widehat{x}_N) \triangleq (\rho_d - C\widehat{\xi}_N)^2 = U_k^T \widehat{\Phi}_k^T C^T C \widehat{\Phi}_k U_k - 2\widehat{\lambda} C \widehat{\Phi}_k U_k + \widehat{\lambda}^T \widehat{\lambda} \tag{A.5}$$

$$\widehat{A} \triangleq A(\widehat{\omega}_k);\ \widehat{B} \triangleq B(\widehat{\beta}_k, \widehat{\omega}_k) \tag{A.6}$$

$$\widehat{\Phi}_k \triangleq [\widehat{A}^{N-k-1}\widehat{B}, \ldots, \widehat{A}^1\widehat{B}, \widehat{B}];\ \widehat{\Psi}_k \triangleq \widehat{A}^{N-k} \tag{A.7}$$

$$\widehat{x}_k = \begin{bmatrix} \widehat{\omega}_k \\ \widehat{\beta}_k \\ \widehat{\xi}_k \end{bmatrix};\ \widehat{\lambda} \triangleq \rho_d - \widehat{\Psi}_k \widehat{\xi}_k \tag{A.8}$$

$$\rho_d \triangleq \text{Desired position at stage } N \tag{A.9}$$

The cost (A.5) is computed for each of the 2^{N-k} enumerated control sequences U_k. The one with smallest cost is denoted as the optimal sequence U_k^*,

$$U_k^* \triangleq \arg\min_{U_k} g_N(\widehat{x}_N) \tag{A.10}$$

$$U_k^* = [u_k^*, u_{k+1}^*, \ldots, u_{N-1}^*]^T \tag{A.11}$$

The first component u_k^* of U_k^* is defined as the HCE control at time k,

$$u_k^{HCE} = u_k^* \tag{A.12}$$

B APPENDIX: OLF Control

The OLF control at time k for a terminal cost problem is calculated by minimizing the expected cost,

$$\min_{U_k} E[g_N(x_N)|I_k] \tag{B.1}$$

where the controls being optimized over are given by the open-loop sequence,

$$U_k = [u_k, u_{k+1}, \ldots, u_{N-1}]^T \tag{B.2}$$

$$u_n = \{+1, -1\},\quad n = k, \ldots, N-1 \tag{B.3}$$

and where the terminal expected cost is given as,

$$E[g_N(x_N)|I_k] \triangleq E[(\rho_d - C\xi_N)^2] \tag{B.4}$$

$$\rho_d \triangleq \text{Desired position at stage } N \tag{B.5}$$

For OLF control determination, the cost (B.4) is evaluated using a Monte Carlo approximation,

$$E[g_N(x_N)|I_k] \simeq \frac{1}{s_{OLF}} \sum_{j=1}^{s_{OLF}} (\rho_d - C\xi_N^j)^2 \tag{B.6}$$

A particle filter is used to evaluate the realizations in (B.6), where s_{OLF} is the number of particles (assumed sufficiently large). Specifically, the current particle state $\Omega_k\{x_k^j\}_{j=1}^{s_{OLE}}$ at time k is propagated without measurement (i.e., open-loop) from time k to time N for each of the 2^{N-k} enumerated control sequences U_k. The one with smallest cost is denoted as the optimal sequence U_k^*,

$$U_k^* \triangleq \arg\min_{U_k} E[g_N(x_N)|I_k] \tag{B.7}$$

$$U_k^* = [u_k^*, u_{k+1}^*, \ldots, u_{N-1}^*]^T \tag{B.8}$$

The first component u_k^* of U_k^* is defined as the OLF control at time k,

$$u_k^{OLF} = u_k^* \tag{B.9}$$

C APPENDIX: Properties of Stopping Rule

This Appendix discusses properties of the control search stopping rule given by (5.3). Define a decision variable d as,

$$d = J_k(2) - J_k(1) \tag{C.1}$$

where,

$$J_k(i) = E[L_k(i)], \quad i = 1, 2 \tag{C.2}$$

Let the Monte Carlo estimate $\hat{d}$ of d be defined as,

$$\widehat{d}(m)=\widehat{J}_k(2)-\widehat{J}_k(1) \tag{C.3}$$

$$\widehat{J}_k(i)=\frac{1}{m}\sum_{n=1}^{m}L_k^n(i),\quad i=1,2 \tag{C.4}$$

where m MC trajectories are used in the calculation. The decision variable d is estimated by $\hat{d}$ with asymptotically Normal statistics,

$$p(d|m\ \text{measurements})=N(\widehat{d}(m),\sigma_d^2(m)) \tag{C.5}$$

The following discussion will assume asymptotic statistics, where $\sigma_d^2(m)$ is tentatively assumed known. Let a stopping rule $\mathcal{T}$ based on $\hat{d}$ be defined according to (5.3) as,

$$\mathcal{T}(m)\triangleq\begin{cases}\text{stop} & \text{if}|\widehat{d}(m)|+\delta J\geq\alpha\sigma_d(m)\\ \text{continue} & \text{otherwise}\end{cases} \tag{C.6}$$

Let a control decision rule $\mathcal{D}$ based on $\hat{d}$ be defined as,

$$u_k=\mathcal{D}(m)\triangleq\begin{cases}1 & \text{for}\ \widehat{d}(m)>0\\ -1 & \text{for}\ \widehat{d}(m)\leq 0\end{cases} \tag{C.7}$$

Let an event $\mathcal{E}$ be defined as,

$$\mathcal{E}\triangleq\{\text{Event that a control is applied having an associated expected cost} \tag{C.8}$$

$$\text{greater than}\ \delta J\ \text{units larger than the optimal}\} \tag{C.9}$$

LEMMA C.1

Let the search process be terminated using stopping rule $\mathcal{T}$, at which time the control is determined by decision rule $\mathcal{D}$. Then

$$p(\mathcal{E}|\mathcal{D},\mathcal{T})\leq\gamma \tag{C.10}$$

where,

$$\gamma = .1587 \; for \; \alpha = 1 \tag{C.11}$$

$$\gamma = .0227 \; for \; \alpha = 2 \tag{C.12}$$

$$\gamma = .0013 \; for \; \alpha = 3 \tag{C.13}$$

Proof

$$\begin{aligned} p(\mathcal{E}|\mathcal{D}, \mathcal{T}) = & \, p\left(\mathcal{E}|\mathcal{D}, \mathcal{T}, \widehat{d}(m) > 0\right) p\left(\widehat{d}(m) > 0\right) \\ & + p\left(\mathcal{E}|\mathcal{D}, \mathcal{T}, \widehat{d}(m) \leq 0\right) p\left(\widehat{d}(m) \leq 0\right) \end{aligned} \tag{C.14}$$

$$\begin{aligned} = & \, p\left(d \leq -\delta J|\mathcal{D}, \widehat{d}(m) + \delta J \geq \alpha\sigma_d(m), \widehat{d}(m) > 0\right) p\left(\widehat{d}(m) > 0\right) \\ & + p\left(d \geq \delta J|\mathcal{D}, -\widehat{d}(m) + \delta J \geq \alpha\sigma_d(m), \widehat{d}(m) \leq 0\right) p\left(\widehat{d}(m) \leq 0\right) \end{aligned} \tag{C.15}$$

$$\leq \gamma(\alpha\sigma_d) p(\widehat{d}(m) > 0) + \gamma(\alpha\sigma_d)\left(1 - p(\widehat{d}(m) > 0)\right) \tag{C.16}$$

$$= \gamma(\alpha\sigma_d) \tag{C.17}$$

where γ $(\alpha\sigma_d)$ is the probability in the one-sided tail of a Gaussian variate at *a* standard deviations σ_d away from its mean. Values for γ are tabulated in (C.11) (C.12) (C.13). The first term in (C.15) follows from Figure C.1 and evaluation of the stopping rule $\mathcal{T}$ (in (C.6)) on the condition $\hat{d}$ $(m) > 0$. A similar diagram and argument can be made for the second term using the condition $\hat{d}$ $(m) \leq 0$. Equation (C.16) follows from (C.15) by noting from Figure C.1 that the indicated tail area can be overbounded by γ.

In practice, the value for $\sigma_d^2(m)$ is not known exactly. Instead, a value is estimated using the unbiased formula (5.5), and substituted into all relevant expressions.

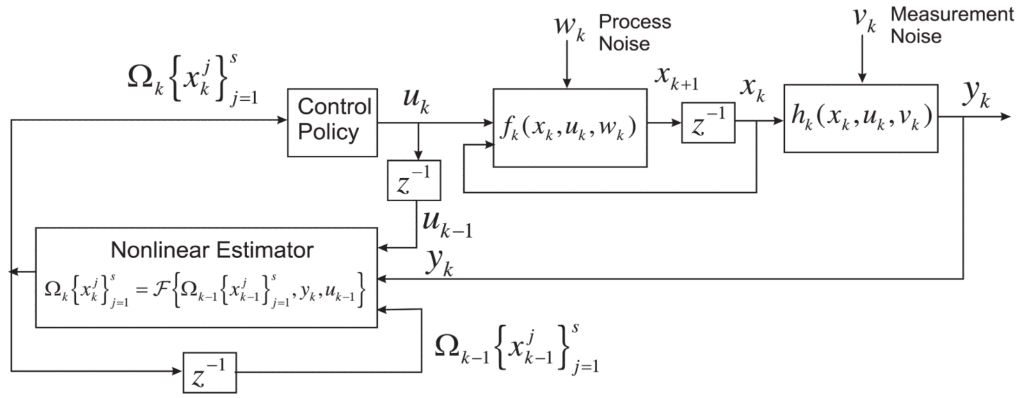

Figure 3.1.
Stochastic control framework based on particle filtering.

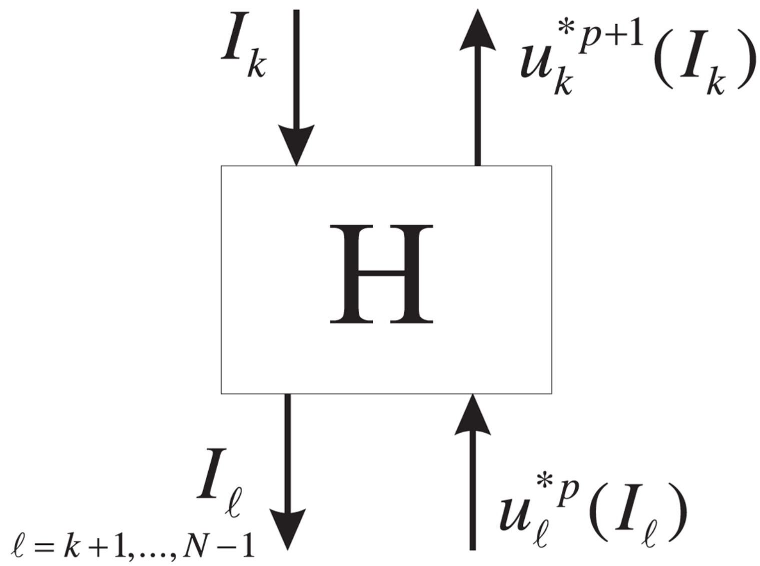

Figure 4.1.
H-Block implementation of a policy iteration.

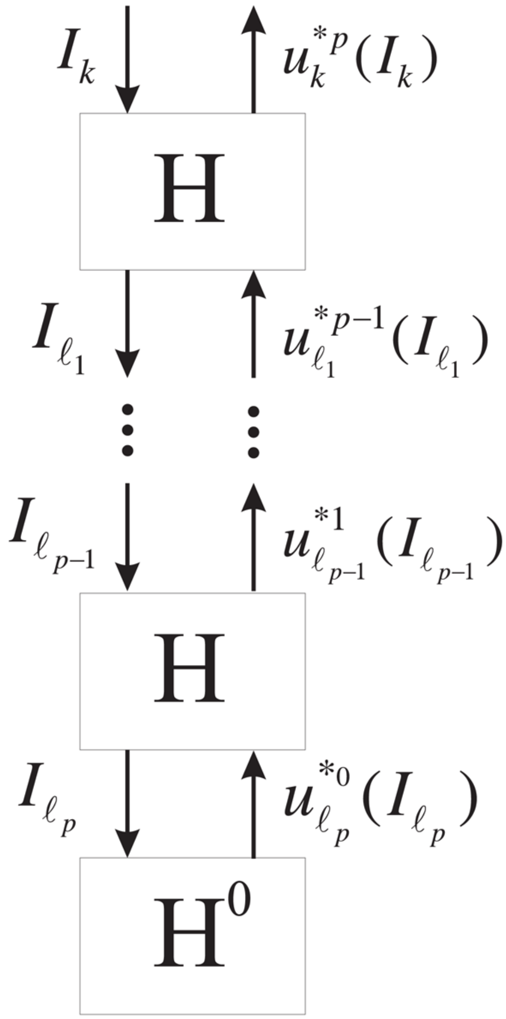

Figure 4.2.
H-Block cascade implementation of multiple policy iterations.

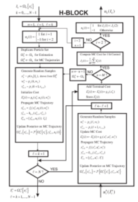

Figure 5.1.
H-block implementation for implicit dual control. Relay type control for simplicity $u = \pm 1$.

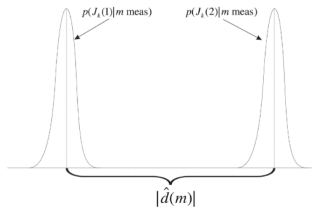

Figure 5.2.
Situation for stopping rule when $|\hat{d}(m)| \gg \delta J$.

NIH-PA Author Manuscript
NIH-PA Author Manuscript

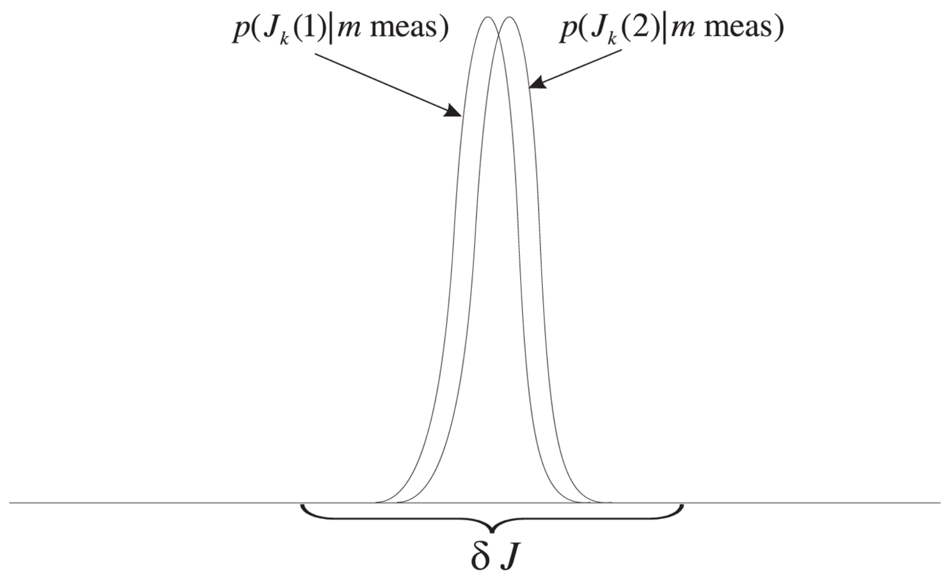

Figure 5.3.
Situation for stopping rule when $|\hat{d}(m)| \lll \delta J$.

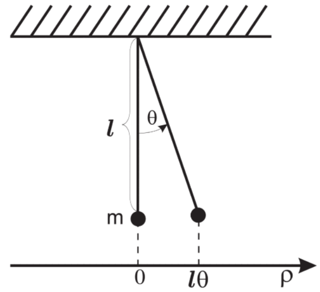

Figure 6.1.
Pendulum model.

NIH-PA Author Manuscript
NIH-PA Author Manuscript

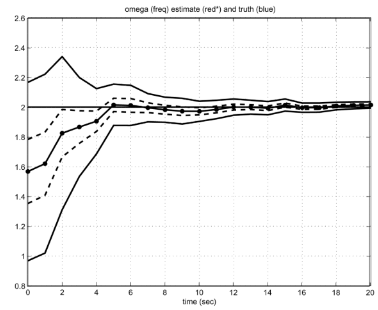

Figure 7.1.
Convergence of pendulum frequency estimate $\hat{\omega}$ to its true value of $\omega^* = 2$ (rad/sec), including 50 (broken line) and 95 (solid) percentile bounds.

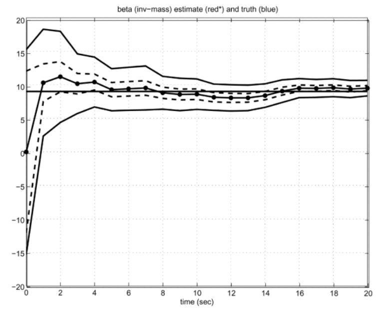

Figure 7.2.
Convergence of |β| to its true value of |β*| = 9 including 50 (broken line) and 95 (solid) percentile bounds.

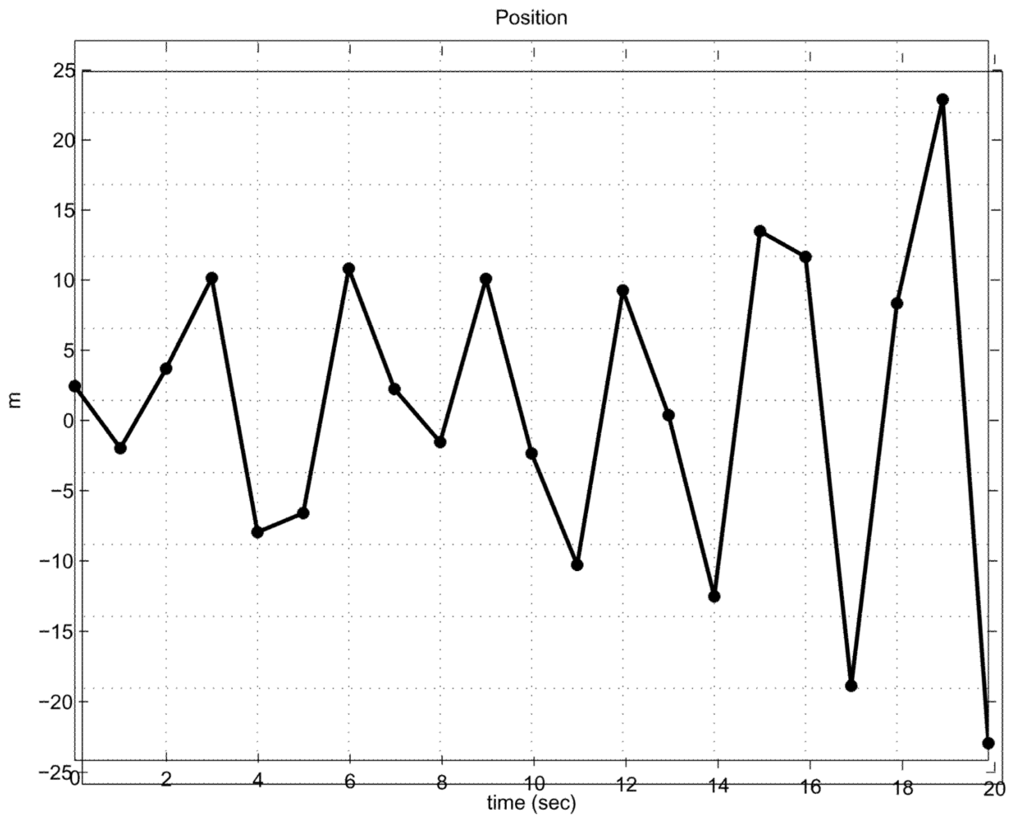

Figure 7.3.
True pendulum position (m).

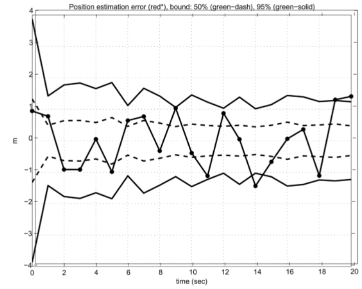

Figure 7.4.
Position estimation error $\rho - \hat{\rho}$ (m) with 50 (broken line) and 95 (solid) percent confidence bounds.

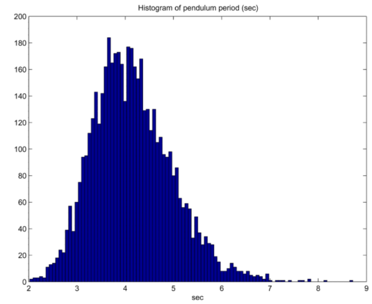

Figure 8.1.
Histogram of pendulum period $\tau = 2\pi/\omega$ (s).

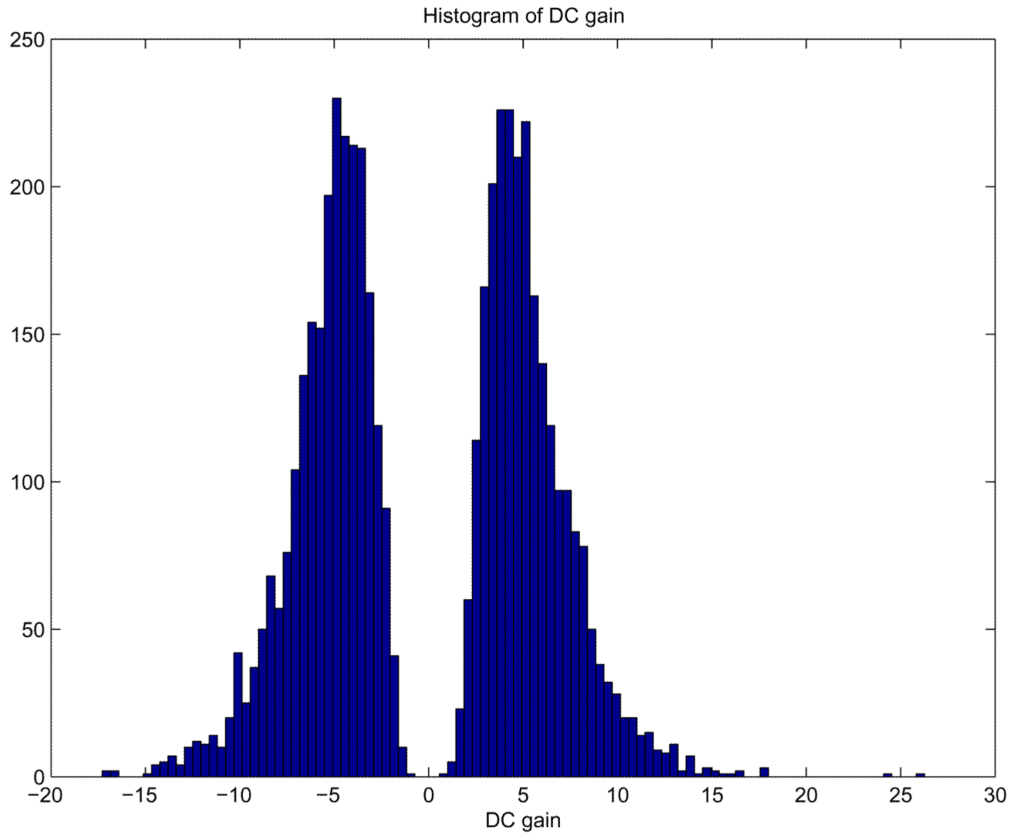

Figure 8.2.
Histogram of pendulum dc gain.

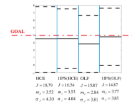

Figure 8.3.

Comparison of Case 2 controller performance results in achieving the goal of 4 (dash-dot line). For each control policy, the mean position m_J (solid) at the final time is shown along with its ±1 standard deviation σ_J (upper and lower dashed line).

NIH-PA Author Manuscript
NIH-PA Author Manuscript

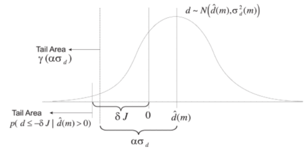

Figure C.1.
Probability of d conditioned on $\hat{d}(m) > 0$.

Table 7.1

Summary of 20 second particle filter run.

Parameter	Truth	Ending Estimate	Starting Estimate	Error	50% Error Bound	95% Error Bound
ω	2	2.0139	1.5677	1.3923e-2	2.1662e-2	3.5500e-2
β	9	9.4964	− 1.4172e-2	4.9638e-1	8.7501e-1	1.6603

NIH-PA Author Manuscript
NIH-PA Author Manuscript

Table 8.2

Summary of Case 1 results.

CASE 1					
Policy	**Cost J**	**# MC Runs**	**MC Error 1σ**	**Mean $m_J = E[\rho_6]$**	**Variability** $\sigma_J = \sqrt{E[(\rho_6 - m_J)^2]}$
HCE	11.087	10,000	0.2060	1.61	3.31
1-IPS(HCE)	8.8968	1,000	0.4363	1.59	2.96

Table 8.3

Summary of Case 2 results.

CASE 2						
Policy	**Cost J**	**# MC Runs**	**MC Error 1σ**	**Mean $m_J = E[\rho_4]$**	**Variability** $\sigma_J = \sqrt{E[(\rho_4 - m_J)^2]}$	**CPU Time (sec)**
HCE	18.794	10,000	0.3460	3.52	4.30	.05
1-IPS(HCE)	16.536	1,000	1.0265	3.53	4.04	20
OLF	15.874	10,000	0.2578	2.84	3.81	.08
1-IPS(OLF)	14.873	1,000	1.1062	3.77	3.85	20