,,

https://doi.org/10.3390/info1010000

MDPI

Article

Closed-Loop Control of Pharmaceutical Manufacturing Processes Can Usually Optimize Better Than Open-Loop Control

Jan Michael Cayabyab Austria[1] and Sarah Marzen[2,*]

[1] Affiliation 1; e-mail@e-mail.com
[2] Affiliation 2; smarzen@natsci.claremont.edu
* Correspondence: e-mail@e-mail.com; Tel.: (optional; include country code; if there are multiple corresponding authors, add author initials) +xx-xxxx-xxx-xxxx (F.L.)

Abstract

Current practice in pharmaceutical batch manufacturing is, in a precise sense, weaker than open-loop control: process parameters are fixed within a validated design space before a batch begins, using settings typically determined by design-of-experiments studies, mechanistic modeling, or engineering judgment during process development. In-process monitoring is limited to anomaly detection against that fixed recipe; a departure triggers a deviation investigation and a batch-disposition decision, not a systematic, data-driven re-optimization of the recipe itself. Data-driven open-loop control that uses artificial intelligence, in which a supervised model is fit to historical batch data and optimized to select recipe settings before each batch begins, is available in the literature but is not yet standard practice. Closed-loop control, in which those settings are further revised during the batch in response to intermediate feedback, is rarer still. This is despite a long-established result in statistical decision theory and control theory that closed-loop (feedback) control weakly dominates open-loop control whenever intermediate information about a system's state is available and properly exploited. We argue that applying this well-known theoretical result to pharmaceutical batch control is a comparatively novel application rather than a novel theoretical claim. We formally state and prove the relevant closed-loop dominance theorem and quantify its practical consequences across three models of pharmaceutical batch control of increasing realism and analytical difficulty: an exactly solvable discrete-state partially observable Markov decision process (POMDP), a closed-form Gaussian model, and a POMDP admitting no closed-form solution. In this last case, we train a stochastic finite-state controller by policy-gradient optimization and compare it against an open-loop baseline. The last of these requires artificial intelligence in both the closed-loop and open-loop scenario. In the first two cases, closed-loop control meaningfully outperforms the open-loop baseline. However, in the last case, we show that this advantage is conditional on adequate optimization: an under-trained closed-loop controller performs no better than, or worse than, a well-optimized open-loop policy, with a sharp rather than gradual transition between the two regimes. We conclude that closed-loop control based on artificial intelligence should be adopted in pharmaceutical manufacturing instead of artificial intelligence-based open-loop control only when enough training samples are present.

Keywords: keyword 1; keyword 2; keyword 3 (List three to ten pertinent keywords specific to the article; yet reasonably common within the subject discipline.)

Received:
Revised:
Accepted:
Published:

Copyright: © 2026 by the authors. Submitted to *Information* for possible open access publication under the terms and conditions of the Creative Commons Attribution (CC BY) license.

1. Introduction

That closed-loop (feedback) control weakly dominates open-loop control whenever intermediate information about a system's state is available and properly exploited is not a new claim: it is a direct consequence of the classical result, due to Blackwell [1] and developed further by DeGroot [2], that the value of information in a Bayesian decision problem is non-negative. Its extension to sequential control problems with partial observability has been standard since Åström [3] reformulated such problems as fully observed Markov decision processes over the belief state and Smallwood and Sondik [4] established the convexity properties of the resulting finite-horizon value function. The practical machinery for exploiting this fact when the optimal closed-loop policy cannot be obtained in closed form– representing policies as finite-state controllers and optimizing them directly, rather than solving the underlying partially observable Markov decision process (POMDP) exactly– dates to Kaelbling et al. [5] and underlies decades of application in robotics, aerospace guidance, and chemical process control [6]. In these domains, the qualitative conclusion that feedback improves on open-loop control is treated as settled engineering practice rather than an open question.

Pharmaceutical manufacturing presents an unusual case in this respect, and it is worth being precise about exactly what is, and is not, currently standard practice. The dominant paradigm for batch manufacturing is not merely open-loop in the sense of Definition 1; it is, in a specific sense, weaker than open-loop control as we use the term throughout this paper. Process parameters are fixed within a validated "design space" before a batch begins [7], with those settings typically determined once, during process development, by design-of-experiments studies, mechanistic modeling, or engineering judgment, rather than by a systematic, data-driven optimization repeated or refined as more batch data accumulates. In-process monitoring is correspondingly limited to *anomaly detection* against this fixed recipe– control charts, multivariate statistical process control, or comparison against a "golden batch" trajectory– and a detected departure triggers a deviation investigation and a batch-disposition decision (continue, quarantine, or reject) rather than any re-optimization of the recipe itself, whether in real time or between batches. This is not a matter of the underlying feedback control theory being unavailable or untested elsewhere; rather, it reflects the cost and regulatory conservatism of validating adaptive, model-based decision-making in a setting where every process change has historically required renewed demonstration of product quality.

A further, data-driven layer above this– fitting a supervised model to historical batch outcomes and optimizing it to select recipe settings before each batch, as in the method of Marzen [8] that we take as a baseline throughout this paper—is available in the literature and is closer to what we call open-loop control in the formal sense of Definition 1, but is likewise not yet standard industrial practice. Even as the U.S. Food and Drug Administration has actively encouraged the adoption of process analytical technology and real-time process understanding [9], and as the industry has begun a slow transition from batch to continuous manufacturing [10], the feedback control actually deployed on the plant floor is typically limited to classical regulation of directly measurable physical variables (temperature, pressure, flow rate) via standard PID loops, rather than closed-loop adaptation of the higher-level process "recipe" in response to inferred, batch-specific quality attributes that are not directly measured. Data-driven open-loop methods such as Marzen [8]'s remain, by construction, unable to revise the resulting settings in light of information that becomes available as the batch proceeds– and it is this further step, from a fixed, anomaly-monitored recipe, to a data-driven open-loop recipe, to a closed-loop one, all potentially using artificial intelligence to maximize their ability to reach optimal reward values, that motivates the present paper.

This gap suggests a comparatively novel application, rather than a novel theoretical claim: the same closed-loop control machinery that is well established elsewhere has, to our knowledge, not yet been systematically applied to the higher-level, recipe-setting layer of pharmaceutical batch control, where the relevant hidden state (e.g., incoming raw-material potency or culture viability) is inferred indirectly rather than measured directly, and where the cost of a poor decision compounds over the batch rather than being correctable instantaneously as in classical low-level regulation. The step we propose is accordingly a step beyond *two* weaker alternatives currently in use or available– a fixed recipe monitored only by anomaly detection, and a data-driven but non-adaptive open-loop recipe such as Marzen [8]'s– rather than a step beyond a single, already-adaptive baseline. The contribution of this paper is accordingly threefold. First, in Section 3, we state and prove the specific form of the closed-loop dominance theorem relevant to this setting, making explicit that it is an instance of established results [1–4] rather than a new theoretical contribution. Second, in Section 4, we quantify the resulting benefit across three models of pharmaceutical batch control of increasing realism and analytical difficulty– an exactly solvable discrete POMDP, a closed-form Gaussian model, and a POMDP with no closed-form solution, solved by training a stochastic finite-state controller– in each case comparing against an open-loop baseline constructed by the method of Marzen [8]. Third, and specific to the practical question of whether this theoretical advantage is realizable in an industry that has been slow to adopt feedback control at this level, we characterize in Section 4.4 how the benefit of closed-loop control depends on the quality of the optimization used to obtain it, showing that the guarantee of Section 3 is a statement about achievable, not automatic, performance, and is therefore conditional on the training and validation effort a manufacturer is willing to commit to a closed-loop implementation.

2. Background

This section reviews the basic definitions of control theory and of partially observable Markov decision processes (POMDPs) needed to state the theoretical result of Section 3 and to interpret the models of Section 4. Readers already familiar with these concepts may proceed directly to Section 3; the definitions given here are standard and are included for self-containedness, following Bellman [11], Puterman [12], and Åström and Murray [13].

2.1. Open-Loop and Closed-Loop Control

At the most basic level, a controlled dynamical system consists of a state that evolves over time, and a control input that an external controller may choose at each time step to influence that evolution. The central distinction relevant to this paper is between two ways a controller may choose its inputs.

Definition 1 (Open-loop control)**.** *A control law is* open-loop *if the control input applied at each time step is a function only of time (and, where relevant, of the system's known initial condition), and does not depend on any measurement or observation of the system taken after the control sequence is fixed. An open-loop controller therefore commits to an entire sequence of control actions in advance and applies that sequence regardless of how the system actually evolves.*

Definition 2 (Closed-loop control)**.** *A control law is* closed-loop *(or* feedback*) if the control input applied at each time step is a function of measurements or observations of the system's behavior obtained up to that time. A closed-loop controller may therefore revise its subsequent actions in response to information that becomes available only after control begins.*

Definitions 1 and 2 are standard in the control theory literature [13] and are typically illustrated with simple examples: a sprinkler system that runs for a fixed duration each day

regardless of soil moisture is open-loop, whereas a thermostat that adjusts heating based on a measured temperature is closed-loop. The central engineering motivation for closed-loop control is that it can compensate for disturbances, model uncertainty, or unknown initial conditions that cannot be accounted for at design time, whereas an open-loop controller must commit to a single course of action that performs well only on average across the uncertainty it cannot observe.

2.2. Markov Decision Processes

Markov decision processes (MDPs) formalize sequential decision-making under uncertainty when the system's state is fully observed at every decision epoch [11,12].

Definition 3 (Markov decision process)**.** *A finite-horizon Markov decision process (MDP) is a tuple (S, A, P, R, T) where S is a finite set of states, A is a finite set of actions, $P : S \times A \times S \to [0, 1]$ is a transition kernel with $P(s' \mid s, a)$ denoting the probability of transitioning to state s' given that action a is taken in state s, $R : S \times A \to \mathbb{R}$ is a bounded reward function, and $T \in \mathbb{N}$ is the decision horizon.*

Definition 4 (Policy and value function)**.** *A (closed-loop) policy for an MDP is a sequence $\pi = (\pi_1, \ldots, \pi_T)$ of functions $\pi_t : S \to \Delta(A)$ mapping the current state to a distribution over actions. The value of a policy π from state s at stage t is*

$$V_t^{\pi}(s) = \mathbb{E}_{\pi}\Big[\textstyle\sum_{\tau=t}^{T} R(s_\tau, a_\tau) \;\Big|\; s_t = s \Big], \tag{1}$$

and the optimal value function $V_t^{\star}(s) = \max_{\pi} V_t^{\pi}(s)$ satisfies the Bellman optimality equation

$$V_t^{\star}(s) = \max_{a \in A} \Big[R(s, a) + \sum_{s' \in S} P(s' \mid s, a)\, V_{t+1}^{\star}(s') \Big], \tag{2}$$

with $V_{T+1}^{\star} \equiv 0$.

Because the state s_t is directly observed at every stage in an MDP, a policy that maps the current state to an action is automatically closed-loop in the sense of Definition 2: it uses all information available about the system up to time t, since that information is simply s_t itself. The distinction between open-loop and closed-loop control becomes substantive only once the state is not directly observed, which motivates the POMDP framework used throughout this paper.

2.3. Partially Observable Markov Decision Processes

Definition 5 (Partially observable Markov decision process)**.** *A finite-horizon partially observable Markov decision process (POMDP) is a tuple $(S, A, O, P, \Omega, R, T, b_0)$ where S, A, P, R, and T are as in Definition 3, O is a finite set of observations, $\Omega : S \times A \times O \to [0, 1]$ is an observation kernel with $\Omega(o \mid s', a)$ denoting the probability of observing o after taking action a and transitioning to state s', and $b_0 \in \Delta(S)$ is the initial distribution over the (unobserved) starting state [3].*

Unlike in an MDP, the controller in a POMDP never directly observes s_t; it has access only to the history of past actions and observations. The standard approach to this difficulty, due to Åström [3], is to reformulate the POMDP as an MDP over *beliefs*.

Notation 1. *We write $\Delta(S)$ for the set of probability distributions over S, and refer to an element $b \in \Delta(S)$ as a* belief.

Definition 6 (Belief update). *Given a belief $b \in \Delta(S)$, an action $a \in A$, and a subsequently observed $o \in O$, the updated belief $b_{a,o} \in \Delta(S)$ is obtained by Bayesian conditioning,*

$$b_{a,o}(s') = \frac{\Omega(o \mid s', a) \sum_{s \in S} P(s' \mid s, a)\, b(s)}{\sum_{s'' \in S} \Omega(o \mid s'', a) \sum_{s \in S} P(s'' \mid s, a)\, b(s)}. \tag{3}$$

Property 1. *For any belief b and action a, the observation-averaged updated belief recovers the belief predicted by the transition kernel alone: $\sum_{o \in O} p(o \mid b, a)\, b_{a,o} = \sum_s P(\cdot \mid s, a)\, b(s)$, where $p(o \mid b, a) = \sum_{s'} \Omega(o \mid s', a) \sum_s P(s' \mid s, a) b(s)$ is the marginal probability of observing o. This is the belief-space analogue of the law of total probability and underlies the convexity argument used in Theorem 2.*

Because the belief b_t is, by construction, a sufficient statistic for the entire history of past actions and observations [3], a POMDP can be treated as an MDP whose state space is $\Delta(S)$ rather than S, with value function $V_t^\star(b)$ satisfying the belief-space Bellman equation

$$V_t^\star(b) = \max_{a \in A} \Big[\mathbb{E}_{s \sim b}[R(s, a)] + \sum_{o \in O} p(o \mid b, a)\, V_{t+1}^\star(b_{a,o}) \Big]. \tag{4}$$

Smallwood and Sondik [4] showed that $V_t^\star$ is a convex, piecewise-linear function of b at every finite horizon t, a fact used directly in the proof of Theorem 2.

2.4. Open-Loop and Closed-Loop Policies in POMDPs

Definitions 1 and 2 can now be stated formally within the POMDP framework used throughout this paper.

Definition 7 (Open-loop policy, POMDP case). *An open-loop policy for a POMDP is a sequence of action distributions $(\pi_1, \ldots, \pi_T)$, $\pi_t \in \Delta(A)$, fixed before the batch begins and applied regardless of any observation realized during execution. Equivalently, it is a closed-loop policy (in the sense of Definition 8) that is constant in the observation history.*

Definition 8 (Closed-loop policy, POMDP case). *A closed-loop policy for a POMDP is a sequence of functions $\pi_t : \Delta(S) \to \Delta(A)$ mapping the current belief—updated via Equation (3) from the realized observation history—to a distribution over actions.*

Example 1. *In the batch-manufacturing setting of this paper, $s \in S$ represents an unobserved, batch-specific quality attribute (e.g., raw-material potency), $a \in A$ represents a control-knob setting, and $o \in O$ represents a noisy intermediate measurement (e.g., an at-line assay reading) taken partway through the batch. An open-loop policy fixes the knob setting for the entire batch using only the prior belief b_0; a closed-loop policy updates the knob setting after observing o, using the resulting posterior belief $b_{a,o}$.*

When the state space, action space, and horizon are small enough, $V_t^\star(b)$ in Equation (4) can be computed exactly (Section 4.1), or, when the belief update takes a conjugate closed form, expressed analytically (Section 4.2). In general, however, exact computation of $V_t^\star$ is intractable, and closed-loop policies must instead be approximated numerically. One standard approach, following Kaelbling et al. [5], is to represent a closed-loop policy as a *finite-state controller*.

Definition 9 (Finite-state controller). *A stochastic finite-state controller with K internal memory nodes for a POMDP is a pair of functions $\big(\mu : \{1, \ldots, K\} \to \Delta(A),\ \eta : \{1, \ldots, K\} \times O \to \Delta(\{1, \ldots, K\})\big)$: μ maps the controller's current internal node to a distribution over actions, and η maps the current node and the most recently observed $o \in O$ to a distribution over the controller's*

next internal node. The controller begins in a fixed initial node and updates its node at every stage using η, without direct access to the belief b_t or the underlying state s_t.

A finite-state controller is closed-loop in the sense of Definition 8, since its chosen action depends on the observation history through the sequence of node transitions, but it need not compute or represent the belief b_t explicitly. When $V_t^\star$ cannot be obtained in closed form, μ and η can instead be optimized directly by simulation-based policy search—for instance, by policy-gradient ascent on the controller's parameters [14]—which is the approach used to obtain the closed-loop policy of Section 4.3.

3. Theoretical Justification: Closed-Loop Control Weakly Dominates Open-Loop Control

This manuscript is, in a sense, an exploitation of a general and well-established fact in a novel area of application: under optimal execution, a closed-loop policy can never do worse in expectation than an open-loop policy, and it does strictly better whenever the intermediate observation is informative about the reward-relevant hidden state. This is a specific instance of the classical result that *the value of information is non-negative*, due to Blackwell [1] and developed further in the statistical decision theory of DeGroot [2]; its extension to sequential, finite-horizon partially observable control problems follows from the convexity of the belief-state value function established by Åström [3] and Smallwood and Sondik [4]. We state both the single-decision-epoch result and its sequential extension, since together they justify the comparisons in both the static (Section 4.1, 4.2) and dynamic (Section 4.3) settings above.

These theorems are actually quite intuitive. The open-loop policy class is a strict subset of the closed-loop policy class– every open-loop policy is a closed-loop policy that happens to ignore all observations– so the dominance follows trivially from optimizing over a strictly larger feasible set [cf. the general principle of optimality, 6]. It is this containment, together with Blackwell's convexity argument for *why* the larger set is exploited, that jointly justify the theorems presented below, but see the theorem proofs for details.

3.1. Single-Epoch Case: Non-Negative Value of Information

Let Θ be a (possibly continuous) space of hidden states, let $b \in \Delta(\Theta)$ denote a belief (probability distribution) over Θ, and let A be a finite action space with bounded reward function $R : \Theta \times A \to \mathbb{R}$. Define the *Bayes value* of a belief b as the expected reward attainable by choosing the best action given only that belief:

$$u(b) \;:=\; \max_{a \in A} \; \mathbb{E}_{\theta \sim b}\big[R(\theta, a)\big]. \tag{5}$$

Theorem 1 (Non-negative value of information; 1, 2)**.** *Let b be a prior belief and let z be any observation with conditional distribution $p(z \mid \theta)$, inducing marginal law $p(z) = \sum_\theta b(\theta)\, p(z \mid \theta)$ and posterior beliefs $b_z(\theta) := p(\theta \mid z)$. Then*

$$\mathbb{E}_z\big[u(b_z)\big] \;\geq\; u(b), \tag{6}$$

with equality if and only if there exists an action $a^\star$ that is simultaneously optimal under b and under b_z for $p(z)$-almost every realization of z.

Proof. Each map $b \mapsto \mathbb{E}_{\theta \sim b}[R(\theta, a)]$ is linear (affine) in b for fixed a, since it is a weighted sum of the constants $R(\theta, a)$ with weights $b(\theta)$. By Eq. (5), u is therefore a pointwise maximum over $a \in A$ of a finite family of affine functions of b, and the pointwise maximum of affine functions is convex. Hence u is convex on $\Delta(\Theta)$.

By the law of total probability, the observation-averaged posterior recovers the prior:

$$\mathbb{E}_z[b_z] \;=\; \sum_z p(z)\, b_z \;=\; \sum_z p(z \mid \theta) b(\theta)\Big|_{\text{marginalized appropriately}} \;=\; b. \tag{7}$$

(This is the standard martingale property of Bayesian updating: the posterior is, on average over the data, equal to the prior.) Applying Jensen's inequality to the convex function u at the random point b_z, and using (7),

$$\mathbb{E}_z[u(b_z)] \;\geq\; u\big(\mathbb{E}_z[b_z]\big) \;=\; u(b), \tag{8}$$

which is exactly (6). Equality in Jensen's inequality holds iff u is affine on the support of the distribution of b_z, which (given u is a max of finitely many affine functions) holds iff a single action attains the maximum in (5) for b and for $p(z)$-almost every b_z. □

The left-hand side of (6) is exactly the value of the *closed-loop* policy that observes z and re-optimizes its action accordingly; the right-hand side, $u(b)$, is exactly the value of the *open-loop* policy, which must commit to a single (possibly randomized) action using only the prior belief b, since by definition it has no access to z at decision time. Thus Theorem 1 states precisely that closed-loop control weakly dominates open-loop control in a single decision epoch, and that the dominance is strict exactly when the optimal action would differ depending on the realized observation.

3.2. Sequential Case: Finite-Horizon POMDPs

The single-epoch result extends to the finite-horizon control problems considered in Section 4 via the belief-state MDP reformulation of a POMDP [3], together with the classical result that the optimal finite-horizon value function is convex in the belief at every stage [4].

Theorem 2 (Closed-loop dominance in finite-horizon POMDPs)**.** *Let a finite-horizon POMDP be given by state space S, action space A, observation space O, horizon T, reward R, transition kernel P, observation kernel Ω, and initial belief b_0. Let $V_t^\star(b)$ denote the optimal expected reward-to-go from belief b at stage t under policies that may condition on the full observation history. Then $V_t^\star(\cdot)$ is convex on $\Delta(S)$ for every $t = 0, \ldots, T$, and consequently*

$$V_0^\star(b_0) \;\geq\; \max_{a_1,\ldots,a_T} \; \mathbb{E}\Big[\textstyle\sum_{t=1}^{T} R(s_t, a_t)\Big], \tag{9}$$

where the right-hand side is the value of the best open-loop *action sequence (fixed before any observations arrive), with equality iff the optimal closed-loop policy never revises its action away from the open-loop-optimal sequence for any realized observation history.*

Proof sketch. Convexity is established by backward induction [4]. At $t = T$ (or, in a discounted infinite horizon, at any stage treated as terminal for the induction), $V_T^\star(b) = \max_a \mathbb{E}_{s\sim b}[R(s,a)]$, which is convex in b by the same argument as in Theorem 1. Inductively, suppose $V_{t+1}^\star$ is convex. Then

$$V_t^\star(b) = \max_{a\in A} \Big[\mathbb{E}_{s\sim b}[R(s,a)] + \mathbb{E}_o\big[V_{t+1}^\star(b_{a,o})\big]\Big], \tag{10}$$

where $b_{a,o}$ is the Bayesian belief update after taking action a and observing o. The first term is affine in b; the second is, by the single-epoch argument (Theorem 1, applied with $u = V_{t+1}^\star$), a convex function of b for each fixed a, since it is again an expectation of a convex function over observation-averaged posteriors whose mean recovers a function affine in

b. Hence $V_t^\star$, as a pointwise maximum over a of functions convex in b, is itself convex, completing the induction.

Given convexity of $V_t^\star$ at every stage, applying Theorem 1 at each decision epoch shows that re-optimizing the belief-conditioned action at every stage (closed-loop control) weakly improves on committing to the belief-averaged (open-loop) action at that stage, for every stage t. Summing this stagewise dominance over $t = 1, \ldots, T$ yields (9). □

Remark 1 (The role of "doing everything right")**.** *In practice, a closed-loop controller obtained by approximate optimization is not guaranteed to attain $V_t^\star$, and an under-optimized closed-loop policy can in principle be outperformed by a well-optimized open-loop one. The theorem therefore licenses the claim that closed-loop control is* never worse in principle, *while the empirical comparisons of Section 4 are needed to confirm that the specific closed-loop controller trained there in fact realizes a meaningful fraction of this guaranteed advantage.*

4. Results

To assess the utility of closed-loop control relative to the open-loop approach of [8], we construct three progressively richer models of a pharmaceutical batch process, each instantiating a partially observable Markov decision process (POMDP) in which a batch-specific quality state is unknown at the outset but partially revealed by an intermediate, noisy measurement. This is designed to match, in a rough way, what actually happens in pharmaceutical manufacturing in which cell growth is monitored every so often while the process continues in real time. In each case we compare an open-loop policy– a fixed control-knob setting chosen before the batch begins and never revised—against a closed-loop policy that updates the knob setting in response to intermediate feedback. The three models differ in how the closed-loop policy is obtained: by exact enumeration (Section 4.1), by closed-form Bayesian updating (Section 4.2), and by methods in artificial intelligence when no closed-form solution exists (Section 4.3). While the theorems of Section 3 guarantee that closed-loop control weakly dominates open-loop control methods, it does not guarantee that closed-loop control *implementations* always weakly dominate open-loop control implementations, as we shall see in Section 4.4.

4.1. Discrete Three-State POMDP: Exact Solution

We first consider a minimal POMDP with three hidden states $s \in \{H, M, L\}$, representing high, medium, and low incoming raw-material or feedstock quality, drawn uniformly at the start of each batch. At two decision points the controller selects a knob aggressiveness $a \in \{a_H, a_M, a_L\}$, with per-stage reward $R(s, a)$ given by

$$R(s,a) = \begin{cases} 100 & \text{if } a \text{ matches } s, \\ 60 & \text{if } a \text{ is one level off,} \\ 20 & \text{if } a \text{ is two levels off (opposite extremes).} \end{cases} \tag{11}$$

Between the two decisions, the controller receives one noisy observation $o \in \{H, M, L\}$ of the true state, correct with probability q and otherwise uniform over the remaining two states.

The open-loop policy commits to a single action for both stages before any observation is available; by symmetry the optimal fixed choice is the compromise action a_M, yielding expected total reward 146.7. The closed-loop policy chooses $a_1 = a_M$ (no information yet available) and then updates its second action via exact Bayesian belief updating given o, switching to the action matching the most probable state whenever $q > 0.5$. As an example,

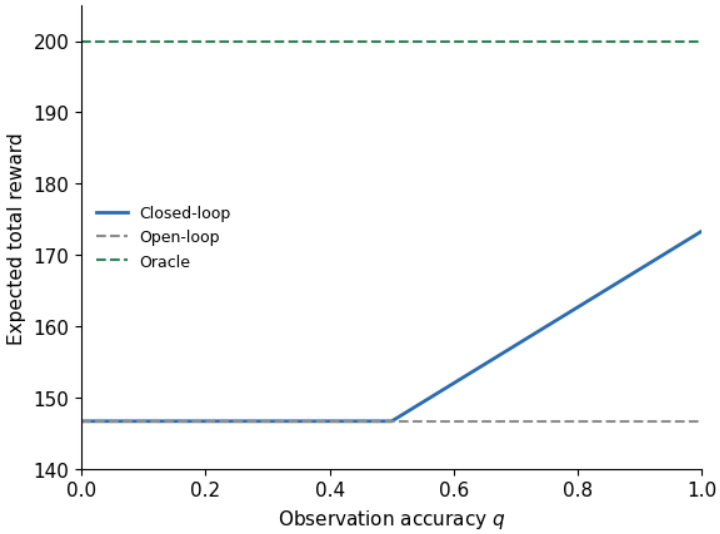

Figure 1. How open-loop reward compares to closed-loop reward as the informativeness of an intermediate observation, parametrized by q, varies. The intermediate observation modifies the closed-loop controller but not the open-loop controller. For $q \leq 0.5$, the intermediate observation provides no information, and closed-loop and open-loop control are identical. But for $q > 0.5$, the intermediate observation is informative, and closed-loop control weakly dominates open-loop control. An oracle that knows exactly the hidden state dominates both controllers.

with $q = 0.7$, this yields expected total reward 157.3, compared with 200.0 for an oracle policy that observes the true state directly.

Notably, the value of closing the loop in this model is governed by a sharp threshold: closed-loop control outperforms open-loop control if and only if $q > 0.5$, i.e., whenever the intermediate observation is even marginally more informative than chance. Below this threshold, the noisy observation is uninformative enough that the open-loop compromise action remains optimal even after the observation is revealed, and theorem weakly guaranteeing improvement only actually guarantees equality or better. Figure 1 summarize these results.

4.2. Gaussian Model: Closed-Form Continuous Solution

We next consider a continuous generalization in which the reward structure of [8]'s own toy example, $y = x^2 + \eta$ (optimal at $x^* = 0$), is extended to include an unknown, batch-specific shift θ in the optimum:

$$y = (x - \theta)^2 + \eta, \qquad \theta \sim \mathcal{N}(0, \sigma_0^2), \qquad \eta \sim \mathcal{N}(0, \sigma_\eta^2). \tag{12}$$

Applying the open-loop method of [8] directly to historical data pooled across many batches (i.e., across the distribution of θ) recovers exactly $x^* = 0$, since the fitted model learns the marginal relationship $\mathbb{E}_\theta[(x - \theta)^2] = x^2 + \sigma_0^2$. This value of x^* is optimal on average across batches but not for any particular batch's realized θ.

The closed-loop policy instead uses a single mid-batch measurement $z = \theta + \varepsilon$, $\varepsilon \sim \mathcal{N}(0, \sigma_z^2)$, and sets the second-stage knob to the Bayesian posterior mean $x_2 = \kappa z$, where

$$\kappa = \frac{\sigma_0^2}{\sigma_0^2 + \sigma_z^2} \tag{13}$$

is the Kalman gain. This yields expected total cost

$$C_{\text{closed-loop}} = \sigma_0^2(2 - \kappa) + 2\sigma_\eta^2, \qquad C_{\text{open-loop}} = 2\sigma_0^2 + 2\sigma_\eta^2, \tag{14}$$

so that closed-loop control reduces expected cost by exactly $\sigma_0^2\kappa = \sigma_0^4/(\sigma_0^2+\sigma_z^2)$ for *any* finite sensor noise σ_z^2—in contrast to the sharp threshold of Section 4.1, the benefit here degrades smoothly and vanishes only in the limit of an uninformative sensor ($\sigma_z^2 \to \infty$). Figure 2 illustrates this for representative parameters $\sigma_0^2 = 1$, $\sigma_\eta^2 = 0.5$.

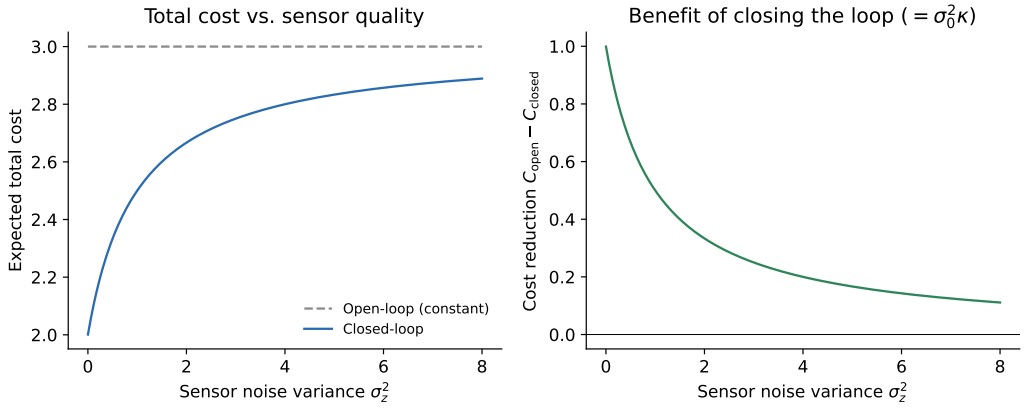

Figure 2. Left: expected total cost for open-loop and closed-loop policies as a function of sensor noise variance σ_z^2 ($\sigma_0^2 = 1$, $\sigma_\eta^2 = 0.5$). Right: the resulting cost reduction from closing the loop, $\sigma_0^2\kappa$, which decreases smoothly rather than exhibiting the threshold behavior of the discrete model.

4.3. Finite-State Controller: A Non-Analytically-Solvable POMDP

Finally, we consider a POMDP for which neither exact enumeration nor a closed-form Bayesian update is available. The hidden state $s \in \{0,1,2,3\}$ represents a batch-quality regime that degrades stochastically over $T = 5$ stages, with degradation probability increasing in the aggressiveness of the chosen action– so that aggressive settings extract more reward when the state is good but accelerate the transition to states in which aggressive settings are actively harmful (Table 1). A noisy observation $o \in \{0,1,2,3\}$ of the resulting state (55% accurate) is available before each subsequent decision (Table 2). Transitions are based on the transition probabilities in Table 3.

Table 1. Reward $R(s,a)$ in the non-analytically-solvable POMDP.

State s	Gentle	Moderate	Aggressive
0 (best)	5	8	12
1	4	6	5
2	3	3	1
3 (worst)	2	1	−3

Because the chosen action affects both immediate reward and the transition dynamics of the hidden state, the optimal closed-loop policy cannot be obtained by a single Bayesian update as in Section 4.2; it must instead be found by numerical policy search. We represent the closed-loop policy as a stochastic finite-state controller (FSC) following [5], with $K = 4$ internal memory nodes: each node specifies a distribution over actions, and each (node, observation) pair specifies a distribution over successor nodes. The controller parameters are trained by policy-gradient ascent (REINFORCE with an Adam update rule), following the general approach of [14] for optimizing stochastic finite-state controllers when exact POMDP planning is intractable.

For the open-loop baseline, we apply the method of [8] exactly as published: a fixed action distribution (the "knob setting") is chosen once per batch and held constant for all T stages; noisy single-batch outcomes are generated across a range of such settings; a neural

Table 2. Observation kernel $\Omega(o \mid s)$ used in the non-analytically-solvable POMDP of Section 4.3. Each row gives the distribution over observations o conditional on the true (resulting) state s; rows sum to 1. The diagonal entries are fixed at 0.55, with the remaining probability mass distributed over the incorrect observations in inverse proportion to their distance from the true state, so that neighboring states are more likely to be confused for one another than distant ones.

	o			
s	0	1	2	3
0	0.5500	0.2455	0.1227	0.0818
1	0.1800	0.5500	0.1800	0.0900
2	0.0900	0.1800	0.5500	0.1800
3	0.0818	0.1227	0.2455	0.5500

Table 3. Transition kernel $P(s' \mid s, a)$ used in the non-analytically-solvable POMDP of Section 4.3. For each action, the state either remains unchanged (probability stay(a)) or degrades by exactly one level (probability $1 - \text{stay}(a)$), with $\text{stay}(a) = 0.85, 0.70, 0.50$ for gentle, moderate, and aggressive actions, respectively; more aggressive actions therefore degrade the state faster. State $s = 3$ (worst) is absorbing under every action. Omitted entries are 0.

		s'			
Action a	s	0	1	2	3
Gentle (stay = 0.85)	0	0.85	0.15	—	—
	1	—	0.85	0.15	—
	2	—	—	0.85	0.15
	3	—	—	—	1.00
Moderate (stay = 0.70)	0	0.70	0.30	—	—
	1	—	0.70	0.30	—
	2	—	—	0.70	0.30
	3	—	—	—	1.00
Aggressive (stay = 0.50)	0	0.50	0.50	—	—
	1	—	0.50	0.50	—
	2	—	—	0.50	0.50
	3	—	—	—	1.00

network is fit to predict total reward from the setting; and bounded optimization is used to select the setting that maximizes the fitted surrogate.

Table 4 and Figure 3 report the resulting policy values, each evaluated over 3×10^5 independent simulated batches. The trained closed-loop FSC recovers approximately 28% of the gap between the open-loop policy and the fully-observable oracle (computed by exact backward induction), despite having access to only a single, imperfect observation per stage and a controller with minimal memory ($K = 4$ nodes).

Table 4. Expected total reward, non-analytically-solvable POMDP.

Policy	Expected total reward
Open-loop (NN surrogate + optimization, à la [8])	17.72
Closed-loop (trained finite-state controller, $K = 4$)	18.72
Oracle (fully observable, exact backward induction)	21.32

4.4. Practical Considerations: Sensitivity to Training Quality

The result of Section 4.3 assumes the closed-loop controller is trained to convergence. In practice, a firm adopting closed-loop control incurs a real, ongoing computational and

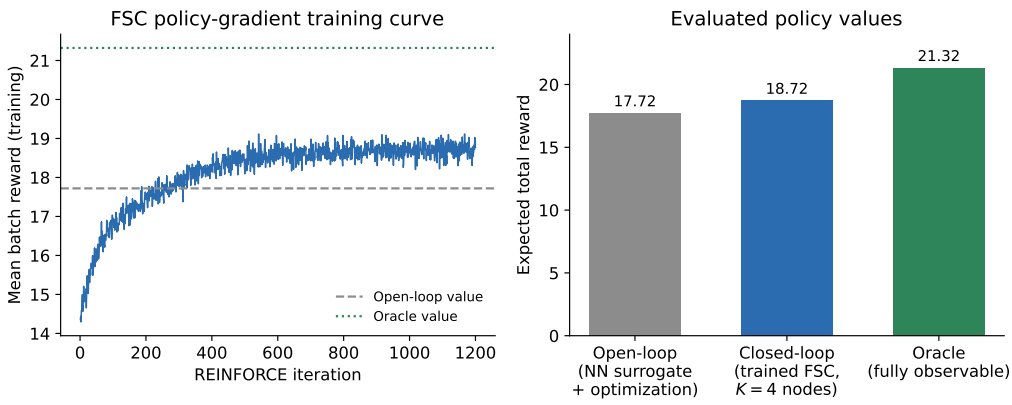

Figure 3. Left: REINFORCE training curve for the finite-state controller (mean batch reward per training iteration), converging toward a plateau above the open-loop value. Right: final expected total reward for the open-loop, closed-loop, and oracle policies, each evaluated over 3×10^5 simulated batches.

engineering cost to train and validate the controller, and this cost may not always be paid in full. We therefore repeated the training procedure of Section 4.3 across a range of substantially reduced training budgets—fewer REINFORCE iterations, smaller batch sizes per iteration, or both—to characterize how the comparison between closed-loop and open-loop control degrades under realistic, imperfect training. For each budget, we trained the finite-state controller from multiple independent random initializations and evaluated each resulting policy's true expected reward via 2×10^4 independent simulated batches, then computed the fraction of training runs whose resulting closed-loop policy outperformed the fixed, well-optimized open-loop baseline of Section 4.3 (17.72). Table 5 and Figure 4 report the results.

Table 5. Closed-loop performance as a function of training budget. Win rate is the fraction of independently trained controllers (out of the number of seeds shown) whose evaluated expected reward exceeded the open-loop baseline of 17.72.

Training tier	Iterations × batch	Seeds	Mean value	Std.	Win rate
Untrained (random init)	—	50	13.63	0.57	0%
Very poor	10×50	50	14.11	0.52	0%
Poor	30×100	50	15.18	0.35	0%
Light	60×150	50	15.79	0.25	0%
Mediocre	150×300	60	16.98	0.14	0%
Moderate	400×800	30	17.72	0.18	46.7%
Well-trained (Sec. 4.3)	1200×4000	15	18.70	0.13	100%

Two features of this result are practically significant. First, the transition is not gradual: below a clear budget threshold, closed-loop control essentially never outperforms open-loop control (0% win rate through the "Mediocre" tier), rather than winning by a smaller but still positive margin. Second, the transition through parity is comparatively narrow—the "Moderate" tier, at roughly a third of the well-trained budget, already produces a win rate statistically indistinguishable from a coin flip ($46.7\% \pm 9.1\%$)—after which further training rapidly and reliably restores closed-loop dominance. Undertrained controllers are also markedly less consistent from run to run (standard deviation falling from 0.57 to 0.13 as training improves), so an organization operating near the transition risks not only a coin-flip outcome on average, but substantial run-to-run variability in which policy it actually ends up deploying.

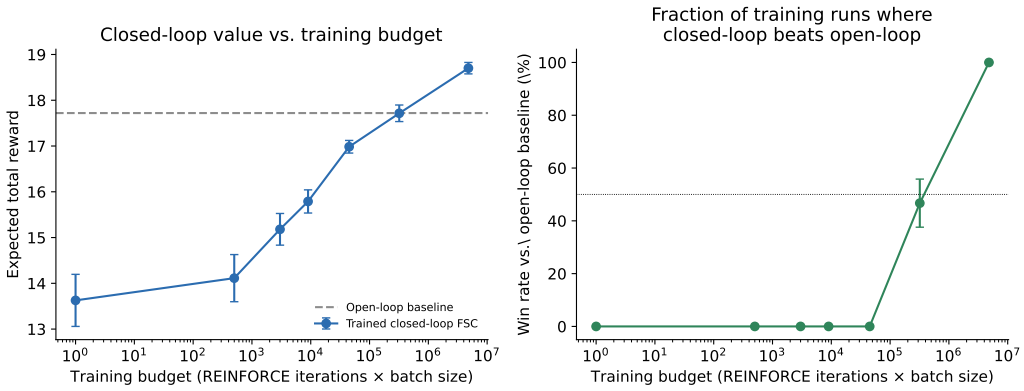

Figure 4. Left: expected total reward of the trained closed-loop policy as a function of training budget (log scale), relative to the fixed open-loop baseline (dashed). Right: fraction of independently trained controllers, at each budget, whose evaluated value exceeded the open-loop baseline, with binomial standard-error bars.

This is precisely the scenario anticipated in Remark 1: Theorem 2 guarantees only that the *achievable optimum* of closed-loop control weakly dominates open-loop control, not that any specific, resource-constrained training run will attain it. For a pharmaceutical manufacturer evaluating whether to adopt closed-loop control, this result implies that the value of the switch is contingent on committing sufficient training and validation effort to the closed-loop controller; a rushed or under-resourced implementation risks realizing no benefit at all relative to a well-optimized open-loop procedure, or even a modest regression, despite the underlying theoretical guarantee that closed-loop control cannot in principle do worse.

Taken together, the four results demonstrate that the benefit of closed-loop control over the open-loop method of [8] persists across settings of increasing realism and analytical difficulty—exact and threshold-governed in the discrete case (Section 4.1), exact and smoothly-varying in the Gaussian case (Section 4.2), and empirically demonstrable via numerical policy search when no closed-form solution exists at all (Section 4.3)—while Section 4.4 shows that realizing this benefit in practice is conditional on the closed-loop controller actually being trained to a sufficient standard, a practical caveat that any deployment plan should budget for explicitly.

5. Discussion

The theoretical underpinnings of this paper are not new. Closed-loop control weakly dominates open-loop control whenever information becomes available during execution and is properly exploited, and this is a direct consequence of results already established in statistical decision theory and control theory [1–4]. What this paper contributes is: first, a concrete demonstration that this well-known principle translates into a quantifiable, non-trivial benefit across three increasingly realistic models of pharmaceutical batch manufacturing (Section 4), including a setting in which the underlying POMDP has no closed-form solution and both controllers are based on artificial intelligence techniques (Section 4.3); and second, an empirical characterization of the conditions under which this benefit is, and is not, actually realized in practice (Section 4.4).

That having been said, the models of Section 4 are deliberately minimal, and several limitations should be kept in view when extrapolating from them. The discrete and Gaussian models (Sections 4.1–4.2) are stylized to admit closed-form analysis and are not fit to real pharmaceutical process data. In particular, the reward structures, state cardinalities, and noise levels were chosen for analytical transparency rather than empirical calibration.

The finite-state controller model of Section 4.3, while free of a closed-form solution, is similarly a toy instantiation of the qualitative dynamics described in Section 1 (reward depending on matching action aggressiveness to an evolving, partially observed quality state) rather than a model fit to any specific unit operation. The training-budget sweep of Section 4.4 is likewise specific to this model, to the particular finite-state-controller architecture and policy-gradient procedure used to train it, and to the range of budgets tested; the location and sharpness of the transition from "closed-loop never wins" to "closed-loop reliably wins" should not be assumed to transfer quantitatively to other POMDP instances, controller architectures, or training algorithms, even though we expect the qualitative existence of such a transition to be a general phenomenon of any closed-loop policy obtained by approximate optimization.

Still, taken at face value, Sections 4.1-4.3 support a straightforward recommendation for pharmaceutical manufacturers: closed-loop control of batch-level knob settings, using even a single, imperfect intermediate measurement, should outperform the open-loop practice that currently dominates the industry [7,10], and should therefore be adopted in place of methods such as that of Marzen [8] wherever such a measurement is available. This recommendation follows directly from Theorem 2 and is not, on its own, a surprising conclusion– it is the pharmaceutical-manufacturing instance of a principle long taken for granted in other control-intensive industries [6].

The result of Section 4.4, however, substantially qualifies how this recommendation should be operationalized. Theorem 2 is a statement about the value attainable by an *optimally trained* closed-loop policy; it says nothing about the value of a specific policy obtained after a specific, finite amount of training. Our sweep over training budgets showed that this distinction is not a minor technicality: a closed-loop controller trained with a mediocre budget did not perform somewhat worse than the open-loop baseline while still capturing part of the theoretical advantage– it performed no better than open-loop at all across every seed we tested until training reached roughly a third of the budget used in Section 4.3 (Table 5), and even then only matched the open-loop baseline on average, with individual training runs landing on either side of it with roughly equal frequency. Below that point, adopting closed-loop control would not have been a smaller win for a pharmaceutical manufacturer. It would have been a loss, obscured by a theorem that guarantees only that the win is available in principle, not that it will be collected by any particular implementation.

This yields a practical decision rule that we believe is the paper's central applied contribution, and that is easy to state precisely because the underlying theory is already settled: *a manufacturer should adopt closed-loop control only once it can demonstrate, by held-out evaluation against the open-loop alternative, that its specific trained controller actually outperforms that alternative– and should default to a well-optimized open-loop policy, rather than an under-trained closed-loop one, whenever that demonstration is not yet available.* This recommendation is deliberately conservative relative to the usual framing of closed-loop control as an unqualified improvement. It reflects the fact that, in a regulated manufacturing environment, an under-performing closed-loop controller is not merely a missed opportunity but an active regression relative to the validated open-loop procedure it would replace, with the associated cost of a failed batch, a deviation investigation, or a qualification review.

Therefore, it is important for artificial intelligence and control theory applications to manufacturing in general to develop a reliable way to determine, for a specific trained controller, whether it has in fact reached the region in which it outperforms a given open-loop baseline, without requiring the large held-out evaluation budgets used in this paper. Developing such a diagnostic– for instance, a stopping rule based on the variance of policy-gradient updates, or a statistical test with an explicit false-adoption rate for the

"closed-loop beats open-loop" decision, adapted to a regulatory setting where conservative error control is required, is a natural next step. A second direction is to replace the stylized reward and transition structures of Section 4 with models fit to real process data from a specific unit operation, to determine whether the training budget required to reach reliable closed-loop dominance in practice is modest enough to be practical within existing validation timelines, or substantial enough that open-loop control over even just simply anomaly detection would apply for the foreseeable future. Finally, the finite-state-controller representation used here is only one of several standard ways to parameterize an approximately optimized closed-loop policy [5,14]; repeating the training-budget sweep with alternative representations (e.g., a recurrent or memory-augmented neural network policy) would clarify whether the sharpness of the transition observed in Section 4.4 is a property of the underlying control problem or an artifact of the specific controller architecture used to solve it.

The tension identified between theory and practice goes deeper. Although we did not have access to proprietary pharmaceutical manufacturing information on control knobs and readouts for this manuscript, those details are of paramount importance when applying these ideas to actual pharmaceutical manufacturing optimization practices. Some actual controls may affect multiple process parameters simultaneously; some control knobs are discretely supported rather than continuously supported, as in when a bag of buffer is added to solution rather than a continuous amount of buffer; and furthermore, in some systems, there may be a time delay between when a control knob is adjusted and when its effects take place, leading to a decrease in reward [15]. All of these need to be taken into account when developing control theory optimization solutions in practice, in addition to the considerations about amount of training data described in this manuscript. A further consideration is that compliance with GMP and regulatory considerations may require substantial testing, validation, and approval that might lead to additional time delays in practice.

Even so, we believe that closed-loop control theory should find applications in not just pharmaceutical manufacturing, but also other manufacturing arenas in which closed-loop control theory is not already used. More complex controllers are likely to supplant the simple ones suggested here. Likely, the exact workflow and architecture of controllers will be very specific to the industry and type and amount of data, making it nearly impossible to speculate here about what will actually work in practice. But ensembles of controllers may allow for a robust solution that works in the generic case, finding the best solution from amongst a list of controller outputs [16]. For instance, an ensemble method can be used to combine recommendations from an open-loop controller that uses artificial intelligence [8] and several closed-loop controllers that use artificial intelligence so that better results are chosen from the mix of controllers. We leave discovery of such solutions to future work.

6. Patents

This work was originally a provisional patent entitled "Closed-Loop AI Control for Pharmaceutical Manufacturing Optimization".

Supplementary Materials: The following supporting code can be downloaded at this Google Colab.

Author Contributions: Conceptualization, J.M.C.A. and S.M.; methodology, S.M.; software, S.M.; validation, S.M.; formal analysis, S.M.; writing– original draft preparation, S.M.; writing– review and editing, J.M.C.A. and S.M.; visualization, S.M. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: Simulated data is available at this Google Colab.

Acknowledgments: During the preparation of this manuscript/study, the author(s) used Claude, Sonnet for the purposes of establishing results and writing a paper draft. The authors have reviewed and edited the output and take full responsibility for the content of this publication.

Conflicts of Interest: The authors declare no conflicts of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript; or in the decision to publish the results.

References

1. Blackwell, D. Equivalent comparisons of experiments. *The annals of mathematical statistics* **1953**, pp. 265–272.
2. DeGroot, M.H. *Optimal Statistical Decisions*; McGraw-Hill: New York, 1970.
3. Åström, K.J. Optimal control of Markov processes with incomplete state information I. *Journal of mathematical analysis and applications* **1965**, *10*, 174–205.
4. Smallwood, R.D.; Sondik, E.J. The optimal control of partially observable Markov processes over a finite horizon. *Operations research* **1973**, *21*, 1071–1088.
5. Kaelbling, L.P.; Littman, M.L.; Cassandra, A.R. Planning and acting in partially observable stochastic domains. *Artificial intelligence* **1998**, *101*, 99–134.
6. Bertsekas, D. *Dynamic programming and optimal control: Volume I*; Vol. 4, Athena scientific, 2012.
7. ICH Harmonised Tripartite Guideline: Pharmaceutical Development Q8(R2). Technical report, International Council for Harmonisation of Technical Requirements for Pharmaceuticals for Human Use (ICH), 2009. Current Step 4 version, August 2009.
8. Marzen, S. A Simple Open-Loop Control Method for Optimizing Manufacturing Control Knobs Using Artificial Intelligence. *Electronics* **2025**, *14*, 2676.
9. U.S. Food and Drug Administration. PAT—A Framework for Innovative Pharmaceutical Development, Manufacturing, and Quality Assurance. Technical report, U.S. Food and Drug Administration, 2004.
10. Lee, S.; O'Connor, T.F.; Yang, X.; Cruz, C.N.; Chatterjee, S.; Madurawe, R.D.; Moore, C.M.V.; Yu, L.X.; Woodcock, J. Modernizing Pharmaceutical Manufacturing: from Batch to Continuous Production. *Journal of Pharmaceutical Innovation* **2015**, *10*, 191–199.
11. Bellman, R. *Dynamic Programming*; Princeton University Press: Princeton, NJ, 1957.
12. Puterman, M.L. *Markov Decision Processes: Discrete Stochastic Dynamic Programming*; Wiley Series in Probability and Statistics, Wiley: New York, 1994.
13. Åström, K.J.; Murray, R.M. *Feedback Systems: An Introduction for Scientists and Engineers*; Princeton University Press: Princeton, NJ, 2008.
14. Hansen, E.A. Solving POMDPs by searching in policy space. *arXiv preprint arXiv:1301.7380* **2013**.
15. Sawaya, Y.; Issa, G.; Marzen, S.E. Framework for solving time-delayed Markov Decision Processes. *Physical Review Research* **2023**, *5*, 033034.
16. Mehta, P.; Bukov, M.; Wang, C.H.; Day, A.G.; Richardson, C.; Fisher, C.K.; Schwab, D.J. A high-bias, low-variance introduction to machine learning for physicists. *Physics reports* **2019**, *810*, 1–124.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.