Bioprocess Control: A Shift in Methodology Towards Reinforcement Learning

Mariana Monteiro,a\* Cleo Kontoravdia\*

aSargent Centre for Process Systems Engineering, Department of Chemical Engineering, Imperial College London, Exhibition Road, London SW7 2AZ, United Kingdom

m.monteiro21@imperial.ac.uk, cleo.kontoravdi98@imperial.ac.uk

Abstract

The production of monoclonal antibodies in mammalian cells is a highly complex and nonlinear process. The industry standard for control of this process fails to capture its complex dynamics, leading to high batch-to-batch variability. This inherent complexity makes bioprocesses challenging to model purely mechanistically, while the lack of rich experimental datasets and the need for interpretability in control policies further prevent the use of fully data-driven solutions. We propose a hybrid methodology for optimizing the nutrient feeding strategy that leverages reinforcement learning. We train the controller with an off-policy method due to its data efficiency. This methodology offers the advantage of not needing access to perfect state knowledge, but rather learning from partial observations of the state, which allows for improved generalization. The continuous learning abilities of the proposed method ensure adaptability in response to process changes, while the inclusion of a mechanistic model in the environment aids in the interpretability of the learned control actions.

**Keywords**: Biomanufacturing, Reinforcement Learning, Control, Mammalian cells

* 1. Introduction

The global market of biologics reached an annual value of US$359 billion in 2022, with an expected increase to US$1421 billion by 2032 (Precedence Research, 2022). Mammalian cells produce up to 80% of the commercially available therapeutic proteins (e.g., monoclonal antibodies - mAbs), with Chinese Hamster Ovary (CHO) cells being the primary production host (Al-Majmaie et al., 2022). The production of mAbs used in treating cancer and autoimmune diseases is a highly complex and nonlinear process with many correlated variables. The industry standard for control of this process involves PID control of pH, temperature, and dissolved oxygen tension in the production bioreactor. However, these traditional control strategies fail to capture the complex dynamics of the process, leading to high batch-to-batch variability (Aehle et al., 2011).

This is problematic in a highly regulated industry like biopharmaceuticals, which must ensure consistent and safe product quality. The complexity of bioprocesses presents challenges in their mechanistic modelling and simulation. In addition, the absence of sufficient experimental data and the necessity for interpretable control policies restricts the utilization of entirely data-driven methods. Hence, hybrid models encompassing mechanistic and data-driven tools provide a suitable compromise (Narayanan et al., 2020). Yet, it is unclear how best to integrate these two components and how to account for plant-model mismatch that characterizes bioprocesses. To this end, Petsagkourakis et al. (2020) proposed a methodology that leverages a policy gradient algorithm to learn the optimal policy distribution of a ‘cheap’ simulation model, and then a separate transfer learning algorithm that transfers the optimal policy distribution to the real and more expensive simulation model. Mowbray et al. (2021) proposed a methodology that uses Gaussian processes for the offline simulation, and then uses the posterior uncertainty prediction to account for plant-model mismatch. Andersson et al. (2023) leveraged a Deep Deterministic Policy Gradient-method (TD3) to determine the optimal policies of liquid chromatography columns, and inputs random noise to the control actions. These works lay the foundation for applying reinforcement learning to more complex case studies that are representative of real industrial systems.

In this work, we propose a model-based reinforcement learning methodology to optimize the nutrient feeding strategy of mammalian cell culture producing recombinant therapeutic products. In reinforcement learning, an agent interacts with an environment and learns to act through trial and error. By receiving rewards for its actions, the agent develops a control policy based on the experiences gathered from the environment. We define the environment as a kinetic model of the production bioreactor, simulating CHO cell growth and mAb production dynamics. The agent receives only partial state observations of the reactor. The observations include information on the concentration of amino acids, cell density, and antibody production - measurements commonly available offline during manufacturing and online or at-line during process development. The agent acts on the environment by manipulating the feeding strategy. Our results suggest that the proposed methodology can lead to an optimal control policy. By continuously learning from its environment, without having direct access to it, the controller can adjust to changing conditions in real time, supporting optimal CHO cell growth and antibody production. This adaptability reduces inconsistencies in the production process, leading to more reliable and predictable outcomes.

* 1. Methodology

Figure 1 depicts the implemented controller. The workflow structure is as follows:

1. Initiate the environment.

2. Simulate the environment for the duration of one step.

3. Provide the state, st, to the agent and calculate the reward, rt.

4. Perform soft policy iteration

5. Provide action at to the environment.

6. Redo steps 2-5 until the end of the batch.

7. Reset the environment with new initial conditions and parameters. Redo steps 1-6 until the stop criterion is satisfied.



Figure 1 Controller Steps

The stop criterion is met when the training reaches 15000 steps, where each step is an 8-hour period of a batch operation (which lasts 2 weeks).

* + 1. Environment

The simulation model, defined as a gym environment (Brockman et al., 2016), is a kinetic model of a fed-batch reactor describing the material balances of the cells, nutrients, and metabolites. The model, described in detail in Monteiro et al. (2023), was parameterized and validated with experimental data. The state, st, corresponds to the measurable variables of the environment, metabolites, and biomass concentrations (in total, 26 variables). The actions, at, are the variables the agent can manipulate to reach a specific objective. In this case, they are the volume in and out, which correspond to the typical manipulated variables available during manufacturing. Both the action and the observation spaces are defined as continuous (a ‘box’) with lower and upper bounds of [-1,1]. It is advised to scale the bounds as the policy algorithm is based on Gaussian processes (see subsection 2.2). To ensure that the controller receives different process conditions at each new batch, the kinetic parameters and initial conditions are varied randomly ± 5%.

* + 1. Agent

The agent is trained according to the Soft Actor-Critic (SAC) (Haarnoja et al., 2018), an off-policy algorithm capable of handling a continuous action space. For that, we consider our system to follow Markov decision process (MDP) defined by (S, A, p, r), where S is the state-space, A, the action space, p the probability of transition between spaces and r the reward from each transition. The policy is defined by $π$, and the $ρ\_{π}$ the trajectories distributions, with regards to a policy. The RL agent learns the optimal policies by maximizing the expected sum of rewards $\sum\_{t=0}^{T}E\_{(s\_{t},a\_{t})\~ρ\_{π}}\left[r\left(s\_{t},a\_{t}\right)\right]$, while also maximising entropy $αΗ(π\left(s\_{t}\right)$ of the policies.

|  |  |
| --- | --- |
| $$J\left(π\right)=\sum\_{t=0}^{T}E\_{(s\_{t},a\_{t})\~ρ\_{π}}\left[r\left(s\_{t},a\_{t}\right)+αΗ(π\left(s\_{t}\right))\right]$$ | (1) |

$α$ is called the temperature parameter, which tunes the stochasticity of the optimal policy,

SAC is composed of an actor who decides which action to take next, and a critic, who estimates how good that action is. The actor performs the *soft-policy improvement* by computing the policy π, according to the maximum entropy (a proxy to randomness and exploration). This policy is modelled as a Gaussian process. The critic performs *soft-policy evaluation*, by computing the soft Q-values for a given policy. In the used algorithm, two independent Q-functions are modelled as neural networks and trained independently to mitigate positive biases. The actor’s neural network followed the suggested structure (Raffin et. al, 2021) with two linear layers (256 units each), separated by ReLU activation functions, plus a linear layer that outputs the mean of the action distribution and another layer for the standard deviation. The two critic neural networks have three linear layers; the first two have 256 units each with ReLU activations, while the last one outputs a single Q-value.

* + 1. Reward function

The definition of the reward, Rt, is of utter importance. In this context, there are four main incentives:

* i1: Reach a target mAb concentration. This is defined by a normal distribution of mean 4000 mg/L (the target goal for mAb) and a standard deviation of 500 (the allowed threshold). This format allows the agent to see a smooth improvement in reward when it approximates the target, which helps learning.
* i2: Reach the target as fast as possible, without harsh changes too suddenly.
* i3: Maintain the target concentration once achieved.
* i4: Respect maximum reactor volume.
* i5: Do not cause any numerical failure in the model of the environment.

Each of the incentives has an associated coefficient term (coef\_iX), which allows their ranking with regards to its importance. Hence, the reward function is written as the sum of the product of the coefficient with the respective incentive:

|  |  |
| --- | --- |
| $$reward= \sum\_{x=1}^{4}coef\_{i\_{x}}×i\_{x}$$ | (1) |

The coefficients for each of the incentives were defined as follows:

Table 1 Values of the coefficients associated with the reward function incentives.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Coef. i1** | **Coef. i2** | **Coef. i3** | **Coef. i4** | **Coef. i5** |
| 100 | 20 | 100 | 20 | 1000 |

Reaching the target mAb concentration (Coef. i1) provides the maximum reward possible, followed by maintaining that target (Coef. i3). A very harsh penalty is given for causing any numerical failure in the model. This ensures the agent is rewarded for keeping inside the environment’s physical constraints.

* 1. Results
		1. Hyperparameter Optimisation

The hyperparameters of the agent model were initially tuned using a variant of Bayesian Optimization (Tree-structured Parzen Estimator) using the Python package Optuna (Akiba et al., 2019). Following the optimization, during the training phase we manually adjusted certain parameters such as the learning rate and the learning start step, as the agent was learning too fast and was returning suboptimal policies without exploring the entire action space.

* + 1. Training performance

The following figures (2a and 2b) represent mAb concentration and rewards during training. During the initial phase of the training, the agent explored the allowed action space with regards to both action variables. This led to mAb concentration initially being below the target range, followed by being above the target range for most of the training period.

|  |  |
| --- | --- |
| A graph with a red line  Description automatically generated | A green graph with white text  Description automatically generated |
| (a) | (b) |
| Figure 2 Training evolution of (a) mAb and (b) rewards as a function of training steps. The red dashed lines indicate (a) the target range for mAb and (b) indicate the maximum possible reward (120) for the controller to achieve |

The red line in figure 2a indicates the acceptable concentration interval. However, as the reward steadily reduces, mAb concentration increasingly violates the target upper bound. The fed-batch nature of this case study makes it challenging to go back, after violating the upper bound of the target. Hence, the controller had to wait for a batch to end (of simulated duration of 14 days), to try a new policy. A higher production rate is not always desirable given that it might lead to immature mAb glycans, which is undesirable (Jimenez del Val et al., 2016). As such, it is important to train the agent not to violate the target upper bound.

The volume fed to the reactor explored the entire action space in the early stages of the training (figure 3a). As the training progresses, it progressively narrows the window of exploration, settling in the upper region of the action space.

|  |  |
| --- | --- |
| A green line graph with white text  Description automatically generated | A green and white sound wave  Description automatically generated |
| (a) | (b) |
| Figure 3 Training evolution of (a) volume in and (b) volume out as a function of training steps.  |

The volume removed from the reactor (figure 3b) displays a u-shaped behavior, in which the period when sampling was the lowest matches the highest mAb concentration. The volume constraint was respected throughout the training, while always hitting the upper bound. This means that the agent uses volume out to achieve objectives without violating the volume constraint. The training results also suggest to not feed as a percentage of the reactor volume nor to overfeed, both of which are common industrial practice). The proposed feeding strategy might potentially avoid common overfeeding problems such as increases in osmolality, which may hinder cell growth (Alhuthali et al., 2021).

* + 1. Testing performance

|  |
| --- |
| A graph of a graph  Description automatically generated with medium confidence |
| Figure 4 Quantity of mAb produced for different test episodes as a function of batch time |

The testing consisted of simulating the environment with the control policies of the trained agent and evaluating system performance. To ensure validity, the simulated environment continued to have different kinetic parameters and initial conditions randomly chosen within a ±5% range, similar to training. In figure 4, we can observe that mAb concentration reaches the target interval, initially violating the upper bound, and stabilizing at a lower value close to it. During testing, it is also possible to observe an extended stationary phase, which points to a continuous feeding operation.

* 1. Conclusions

This study introduces a novel methodology to incorporate a complex bioprocess model with a state-of-the-art off-policy reinforcement learning algorithm. Our approach enables the integration of mechanistic knowledge with data to develop more effective control strategies. Furthermore, the proposed control approach offers adaptability, which reduces the need for frequent reparameterization of the controller model when culture conditions change.

Due to the heavily regulated nature of the pharmaceutical industry, it is not only important to maximize the objective, but to consistently achieve the target amount, even if suboptimal. This is because high production rates have been correlated with immature glycans, which can reduce the therapeutic efficiency of mAbs. The proposed methodology can thus be used to ensure consistent process performance.

Future work includes systematically finding the coefficients of the reward function that best achieve mAb production through optimization and increase the window of randomness applied to the kinetic parameters of the environment when training.

* 1. References

M. Aehle, A. Kuprijanov, S. Schaepe, R. Simutis, A. Lübbert, 2011, Increasing batch-to-batch reproducibility of CHO cultures by robust open-loop control, Cytotechnology, 63, 1, 41-7.

R. Al-Majmaie, D. Kuystermans, M. Al-Rubeai, 2022, Biopharmaceuticals produced from cultivated mammalian cells, Cell Culture Engineering and Technology: In appreciation to Professor Mohamed Al-Rubeai, 3-52, Cham: Springer International Publishing.

D. Andersson, C. Edlund, B. Corbett et al., 2023, Adaptable control policies for variable liquid chromatography columns using deep reinforcement learning, Sci Rep, 13, 11270.

G. Brockman, V. Cheung, L. Pettersson, J. Schneider, J. Schulman, J. Tang, W. Zaremba, 2016, Openai gym, .

T. Haarnoja, A. Zhou, P. Abbeel, S. Levine, 2018, Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor, International Conference on Machine Learning, PMLR.

I. Jimenez del Val, Y. Fan, D. Weilguny, 2016, Dynamics of immature mAb glycoform secretion during CHO cell culture: An integrated modelling framework, Biotechnology Journal, 11, 610-623.

M. Monteiro, S. Fadda, C. Kontoravdi, 2023, Towards advanced bioprocess optimization: A multiscale modelling approach, Computational and Structural Biotechnology Journal, 21, 3639-3655.

M. Mowbray, P. Petsagkourakis, E.A. Rio-Chanona, R. Smith, D. Zhang, 2021, Safe Chance Constrained Reinforcement Learning for Batch Process Control, Comput. Chem. Eng., 157, 107630.

H. Narayanan, L. Behle, M. F. Luna, M. Sokolov, G. Guillén-Gosálbez, M. Morbidelli, A. Butté, 2020, Hybrid-EKF: Hybrid model coupled with extended Kalman filter for real-time monitoring and control of mammalian cell culture, Biotechnology and bioengineering, 117, 9, 2703–2714.

T. Akiba, S. Sano, T. Yanase, T. Ohta, M. Koyama, 2019, Optuna: A Next-generation Hyperparameter Optimization Framework, KDD.

P. Petsagkourakis, I. O. Sandoval, E. Bradford, D. Zhang, E. A. del Rio-Chanona, 2020, Reinforcement learning for batch bioprocess optimization, Computers & Chemical Engineering, 133, 106649.

A. Raffin, A. Hill, A. Gleave, A. Kanervisto, M. Ernestus, N. Dormann, 2021, Stable-Baselines3: Reliable Reinforcement Learning Implementations, Journal of Machine Learning Research.

S. Alhuthali, P. Kotidis, C. Kontoravdi, 2021, Osmolality Effects on CHO Cell Growth, Cell Volume, Antibody Productivity and Glycosylation, Int J Mol Sci, 2021, Mar 24, 22, 7, 3290.