\*

\*[miao.guo@kcl.ac.uk](mailto:miao.guo@kcl.ac.uk)

Abstract

An economic optimal control problem for biomass fermentation is presented. Building on this, this work contributes a performance comparison between Q-Learning and non-linear model predictive control (MPC) as applied to a simulated biomass fermentation system. Two different scenarios are considered. In the first scenario, it is assumed an accurate model is available. In the second scenario, it is assumed there is parametric mismatch between the model and the true plant. In the second scenario, a simple transfer learning approach is used to improve the performance of the Q-Learning controller, while parameter estimation is used to aid the MPC. Trajectories and performance indicators are presented for both controllers and for both scenarios. It is found that Q-Learning out-performs MPC in the first scenario. In the second scenario, transfer learning is found to significantly improve the performance of the Q-Learning, and to outperform a comparison controller combining MPC with moving horizon estimation (MHE).

**Keywords**: Fermentation, MPC, Q-Learning, Transfer Learning, Mismatch

* 1. Background

Fermentation is an important industrial process, used in the production of food, bio-chemicals, and pharmaceuticals. Fermentation of microbial protein in bioreactors offers an efficient means of converting carbohydrate-based substrates into high protein outputs and is more environmentally sustainable than traditional animal-based protein production across a wide-range of metrics (Good Food Institute 2021). However, maximising the performance of these fermentations through process control is complex (Chai et al. 2022), as the process is non-linear and uncertain, with time-varying behaviour.

Within fermentation, a distinction can be made between biomass fermentation, in which the desired product is microbial biomass, and precision fermentation, in which the desired product is a (primary or secondary) metabolite. Previous works on fermentation process control have tended to focus on precision fermentation, such as continuous fermentation of biofuels (Mohd et al. 2016), and batch fermentation of pharmaceuticals (Ashoori et al. 2009), with a relative lack of work on control of continuous biomass fermentation. More particularly, there has been little comparison of reinforcement learning (RL)-based and model predictive control (MPC) methods in continuous biomass fermentation.

Whereas in precision fermentation, particularly to produce high-value pharmaceuticals, efficiency and productivity are arguably secondary considerations, in continuous biomass fermentation, such as to produce dietary proteins, efficiency and productivity are the key performance indicators, as the mass-specific value of the output is lower. Increasing production capacity and improving process efficiency are therefore an important industry challenge for microbial protein manufacturers. The cost-sensitivity of biomass fermentation to produce protein as a commodity for feed or food motivates interest in developing and applying economic, rather than trajectory-following, control approaches.

* 1. Research Objectives

The research objective is to investigate and compare economic control methods suitable for continuous biomass fermentation. This is a non-linear control problem with an economic objective. In this work, building on Nguang et al. (1998), an optimal control problem for maximising biomass production is formulated for the case of a well-mixed reactor with a single growth-limiting substrate.

The control objective was to maximise the biomass output during a 200-hour window, from a defined, low-biomass concentration starting condition. This reflects industrial practice, in which continuous fermentations are subject to regular re-starts, to avoid problems such as genetic drift. A secondary consideration, the substrate utilization efficiency, was not formally included in the control objective. However, efficiency rates for the different control methods are also reported, as they would also be of interest to an industrial process operator.

Two different cases are considered. In the first case, which is a simpler, more idealized scenario, it is assumed an accurate model of the plant is available. In the second case, which is intended to better reflect the real-world complexities of the control problem, the parameters of the plant are only known approximately, resulting in parametric model-plant mismatch.

* 1. Methodology
     1. System Description

We consider a fixed volume system, with Monod growth kinetics and a single growth-limiting substrate. The state-space is of dimension two (biomass concentration and substrate concentration) and the action space of dimension one (dilution rate). The inlet substrate concentration is considered fixed. This system is described in Equation 1, with values used for the constants also stated. To simulate this system numerically, the SciPy ODEint function was used to integrate the system equations.

|  |  |
| --- | --- |
|  | (1a) |
|  | (1b) |
|  | (1c) |
| = Substrate Concentration, g/L; X = Biomass Concentration, g/L  D = Dilution Rate, /h; Ks = 0.45 g/L Monod Half Saturation Constant  μmax = 0.15 /h, Maximum Growth Rate; Yx = 0.5 g/g, Yield Coefficient |  |

* + 1. Control Methods

The performance of two different control methods, namely economic non-linear model predictive control (NL-MPC), and Q-Learning (QL), was evaluated, initially in the zero-mismatch case, and then in the parametric mismatch case. The control objective in both cases is to maximise the total biomass output over a defined period of time (see Equation 2a), subject to the system dynamics set out in Equation 1.

Non-linear model predictive control (NL-MPC) is an often-employed method which formulates the discrete-time, continuous-action, continuous-state control problem as a non-linear program (NLP). However, the "curse of dimensionality" can present problems when the timescale associated with the best strategy is long relative to the process time; the number of steps ahead the MPC considers, known as the control horizon, determines the dimension of the optimisation problem solved at each step.

QL is a simple reinforcement learning technique, for discrete-state, discrete-action, discrete-time systems, in which a sequence of immediate rewards is used to estimate the long-term pay-off of each state-action pair, with these values stored in a table (see Equation 2cii). The trade-off between long- and short-term interests is controlled by the discount factor, γ. After convergence, the QL agent will usually choose the action with the highest payoff, but during training random actions will be taken more often to explore the system dynamics. For a detailed explanation of the QL algorithm, see Watkins and Dayan (1992).

The objective function used for the MPC was the sum of 3 terms, as defined in Equation 2b: total biomass output during the horizon window, a terminal state bonus, and an actuation switching cost. The Euler method was used to discretise the system dynamics for use in the MPC control problem. This defined a non-linear program (NLP). IPOPT (Wächter & Biegler 2006), a local NLP solver, was used to solve the MPC optimisation problem, through the Pyomo (Hart 2011) interface. The performance of the MPC depends on the control horizon, the control timestep, and on the weighting terms used to reduce actuation switching and to promote long-termism (terminal state weight). Both of these terms were added to improve performance.

The reward used for the Q-Learning was biomass output in one timestep, as defined in Equation 2ci. The exploration rate, ε, used in the QL was linearly decremented from the initial rate to the minimum rate. The SciPy ODEint function was used to integrate the system equations in all cases. The results for MPC and QL are both for the best parameters identified through grid search.

|  |  |
| --- | --- |
|  | (2a) |
|  | (2b) |
|  | (2ci) |
|  | (2cii) |

We then consider the case in which there is mismatch between the model and the true process. MPC is, as its name suggests, a model-based technique. RL is a model-free approach. However, because RL requires many interactions to learn how to control a system well, in practice a numerical model of the target process is required for process control applications, with the model used to train the reinforcement learning agent. Some degree of mismatch between model and plant is inevitable in most systems. Therefore, it is important to consider the impact of mismatch on control systems, and to think about how this mismatch can be accommodated.

In the QL controller, mismatch is addressed using a simple transfer learning (TL) approach. First, the agent is pre-trained on a model over a large number of episodes of 200 h each, split into control timesteps of 2 h. Then the agent is re-trained on the true process during a single “live” run, by increasing the exploration rate, in this case from 0.01 to 0.1. In the MPC, potential mismatch is addressed by parameter estimation, with parameters continually re-estimated using Moving Horizon Estimation, based on (noisy) simulated sensor readings.

* 1. Results
     1. Performance in the Case of Zero-Mismatch

QL can outperform MPC if the discretization used is sufficiently fine. This is primarily because of the duration of the fermentation, which means that the MPC is too short-sighted; extending the MPC horizon further causes the solver to be frustrated by local minima. The equivalent parameter in QL, the discount factor, was set so a reward in 200 h time was valued half as much as a reward now. Discretising each state and action variable into 40 levels allowed the performance of the Q-Learning controller to outperform the MPC. The difference in performance between MPC and Q-Learning was relatively small. However, the highest performance configuration of Q-Learning tested consistently outperformed the highest performance configuration of MPC. The parameters used for MPC can be seen in Table 1, and those used for the Q-Learning can be seen in Table 2. The range of outputs for Q-Learning over 10 iterations of training from scratch was [302.9, 307.8] g/L over 200 h, whereas the output of the MPC was 302.54 g/L. Similarly, the range of substrate efficiencies for the Q-Learning was [87.4, 88.6] %, whereas the MPC efficiency was 83.9 %. State, input, and output trajectories for the two control systems can be seen in Figure 2.

Table 1: Best MPC Controller Configuration in case of Known Plant Parameters

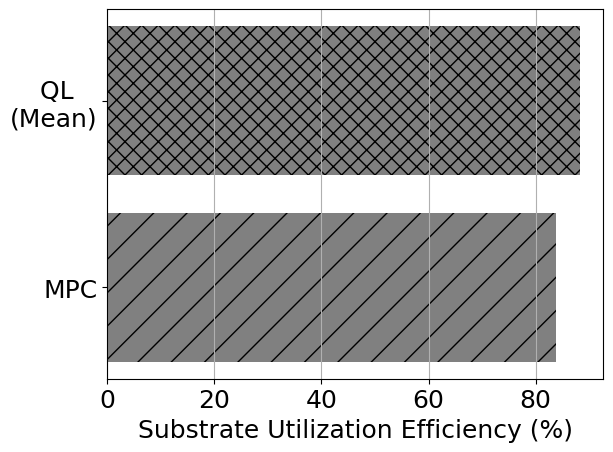
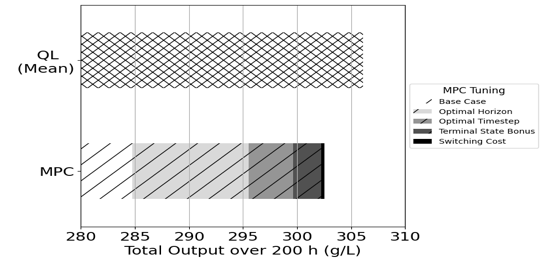


Figure 1: Total Outputs and Substrate Utilization Efficiency for Q-Learning and MPC, with Zero-Mismatch. In the output chart, the impact of the MPC tuning on performance is shown.

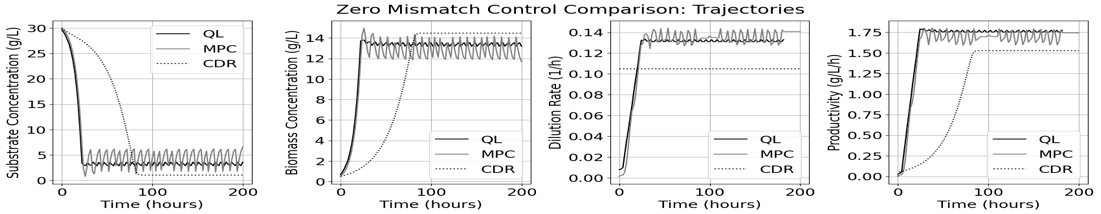


Figure 2: State, Input and Output Trajectories of QL and MPC controlled system, with constant dilution rate (CDR) for reference. Dilution Rate and Productivity plots for QL and MPC have been smoothed with a window of 20 hours.

|  |  |  |  |
| --- | --- | --- | --- |
| Control Timestep, Δt = 2 h | Control Horizon = 44 h | Terminal Weight = 1E-4 | Switch Weight =1E-7 |

Table 2: Best Q-Learning Controller Configuration in case of Known Plant Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| Control Timestep, Δt = 2 h | Initial Exploration Rate, ε0 = 0.5 | Final Exploration Rate, ε0 = 0.01 | Learning Rate, α = 0.3125 |
| Discount Factor, γ =  exp(-Δt ln(2)/Duration) | Number of Discrete Action Levels = 40 | Number of Discrete State Levels per State Variable = 40 | Training Episodes = 128,000 |

* + 1. Transfer Learning

Slight parametric mismatch was applied, by increasing the value of the maximum growth rate, μmax, used in the model from 0.15 to 0.2. Five QL parameters were varied to identify a QL system that performed well when applied to a mismatched system: number of states, number of actions, control timestep, re-training learning rate and re-training exploration rate. The optimum retraining initial exploration rate was 0.1, compared to 0.5 and 0.01 at the start and end, respectively, of pretraining. Better TL results were obtained when the number of state levels (per variable) and actions was reduced to 9 and 18 from 40 and 40, respectively. Increasing the number of learning opportunities during re-training by reducing the control timestep from 2 h to 1 h also improved performance. Both of these later changes come at the expense of performance in a system without mismatch (see Figure 3), which failed to match the performance of the MPC with true parameters. Note that the simpler QL system with true parameters underperforms relative to the MPC system with true parameters.

Using optimal hyperparameters for performance on the target system during the first fermentation episode saw the performance of the RL controller, as measured by output, drop less than 1% when it was pre-trained on a system with a 33% higher maximum growth rate, relative to training on a system without mismatch. Indeed, the QL with TL approach outperformed the optimisation-based approach combining MPC and moving horizon estimation (MHE) for state and parameter estimation in both output and efficiency (see Figure 3). In practice, the re-training hyper-parameters couldn’t be optimised for the target system, but heuristics could be used to estimate the most appropriate re-training hyper-parameters as a function of the scale of the suspected mismatch between the training system and the target system. State, input, and output trajectories for the QL with TL system during its single episode of retraining can be seen in Figure 4.

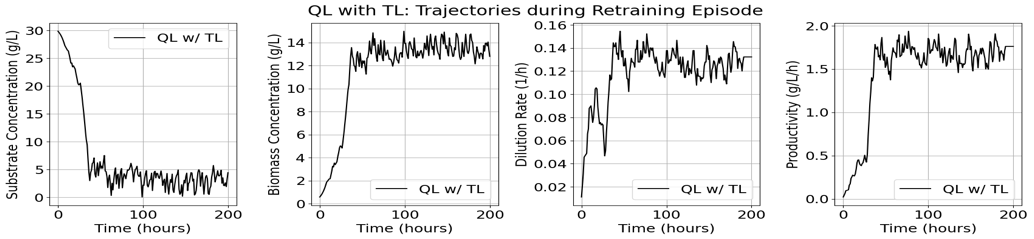


Figure 4: State, Input and Output Trajectories of QL with TL during its retraining episode.

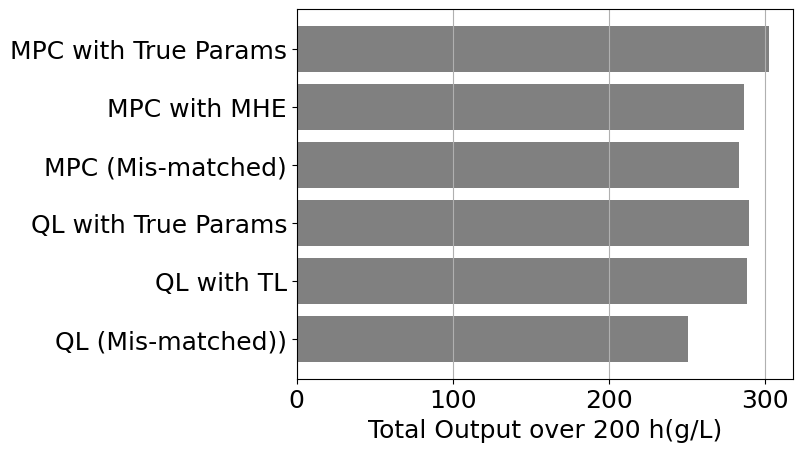
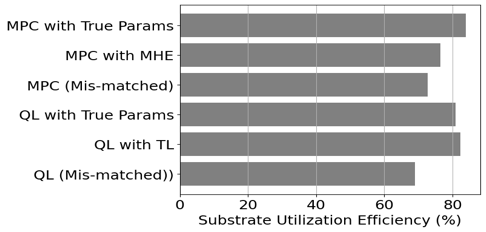


Figure 3: Total Outputs and Substrate Utilization Efficiency for QL and MPC, with slight parametric mismatch.

* 1. Conclusions & Future Work

While QL has been viewed with skepticism by the control community, it is nonetheless a simple control approach that can offer good performance, with QL outperforming a more complex MPC scheme in this application. It should be noted that once the QL controller has been trained and the exploration rate has been reduced to a small value, it is very similar to an explicit MPC controller. As with explicit MPC, for QL the computational effort is concentrated prior to deployment, with considerable training costs, but few online computational costs.

Simple transfer learning can facilitate the rapid adaptation of QL to moderate parametric mismatch. However, there are some trade-offs between designing a small QL system that is quick to adapt to a new system, with mismatch relative to its training system, and maximising the performance of the QL system once it has fully adapted, which may require more complexity. More sophisticated QL schemes that avoid the need to discretize the state and action space could help address this problem, although it is to some extent fundamental.

Future research could investigate RL-MPC hybrids or broaden the controlled variables to include dissolved oxygen, temperature, pH and nitrogen concentration. Additionally, the control systems will be implemented in a real-world setting. This will likely lead to consideration of the effects of non-parametric mismatch. Beyond this, potential future areas of research include high fidelity process simulation for fermentation, application of control approaches to co-culture fermentation systems, and using metabolic modelling, such as flux balance analysis, to derive insights into the control system design.

References

A. Ashoori, B. Moshiri, A. Khaki-Sedigh, and M. R. Bakhtiari, 2009, “Optimal control of a nonlinear fed-batch fermentation process using model predictive approach”, *Journal of* *Process Control*, vol. 19, issue 7, pp. 1162–1173.

W. Y. Chai, K. T. K. Teo, M. K. Tan, and H. J. Tham, 2022, “Fermentation process control and optimization”, *Chemical Engineering & Technology*, vol. 45, issue 10, pp. 1731–1747.

B. Dahhou, G. Roux, and G. Chamilothoris, 1992, “Modelling and adaptive predictive control of a continuous fermentation process”, *Appl. Math. Model.*, vol. 16, issue 10, pp 545-552.

Good Food Institute, 2021 “Fermentation: State of the industry report.”

Hart, W. E., Watson, J. P., and Woodruff, D. L., 2011, "Pyomo: modeling and solving mathematical programs in Python", Math. Program. Comput., vol. 3, issue 3, pp. 219-260.

N. Mohd and N. Aziz, 2016, “Performance and robustness evaluation of nonlinear autoregressive with exogenous input model predictive control in controlling industrial fermentation process”, *Journal of Cleaner Production*, vol. 136, issue 11, pp. 42–50.

S. K. Nguang and X. D. Chen, 1998, “Simple procedure for operating a class of continuous fermentation processes at the optimal steady state productivity”, *Biochemical Engineering Journal*, vol. 1, issue 1, pp. 131-136.

A. Wächter and L. T. Biegler, 2006, “On the Implementation of a Primal-Dual Interior Point Filter Line Search Algorithm for Large-Scale Nonlinear Programming”, *Mathematical Programming* 106(1), pp. 25-57.

Watkins, C.J. and Dayan, P., 1992, “Technical Note: Q-Learning”, *Machine Learning*, vol. 8, pp. 279–292.