BAYESIAN INFERENCE IN PARAMETER ESTIMATION OF BIOPROCESSES

A STUDY ON THE APPLICATIONS OF BAYESIAN INFERENCE TO PARAMETER ESTIMATION PROBLEMS IN BIOPROCESSES

By NIGEL MATHIAS, B. Eng

A Thesis Submitted to the School of Graduate Studies in Partial Fulfillment of the Requirements for the Degree Master of Applied Science

McMaster University

MASTER OF APPLIED SCIENCE (2024)

Hamilton, Ontario, Canada (Chemical Engineering)

TITLE:	A study on the applications of Bayesian inference to parameter estimation problems in bioprocesses
AUTHOR:	Nigel Mathias B. Eng (Chemical Engineering), McMaster University, Hamilton, Canada
SUPERVISOR:	Dr. Prashant Mhaskar
NUMBER OF PAGES:	viii, 65

Abstract

The following thesis explores the use of Bayes' theorem for modelling bioprocesses, specifically using a combination of data-driven modelling techniques and Bayesian inference, to address practical concerns that arise when estimating parameters. This thesis is divided into four chapters, including a novel contribution to the use of surrogate modelling and parameter estimation algorithms for noisy data.

The 2nd chapter addresses the problem of high computational expense when estimating parameters using complex models. The main solution here is the use of surrogate modelling. This method was then applied to a high-fidelity model provided by Sartorius AG. In this, a 3-batch run (simulated) of the bioreactor was passed through the algorithm, and two influential parameters, the growth and death rates of the live cell cultures, were estimated.

The 3rd chapter addresses other challenges that arise in parameter estimation problems. Specifically, the issue of having limited data on a new process can be addressed using historical data, a distinct feature in Bayesian Learning. Finally, the problem with choosing the "right" model for a given process is studied through the use of a term in Bayesian inference known as the evidence. In this, the evidence is used to select between a series of models based on both model complexity and goodness-of-fit to the data.

Acknowledgements

First, I'd like to thank Dr. Prashant Mhaskar for being an excellent, fun, kind and supportive supervisor. I will always be thankful to him for giving me a chance at his research group when I was completing my undergraduate studies at McMaster. His insight and guidance have shaped my research and how I think about our field. I'd also like to thank my other mentor, Dr. Brandon Corbett. His feedback and suggestions have always been brilliant, and I hope he will continue to share his wisdom with me. Next, I'd like to thank the graduate students in our research group, who I now consider lifelong friends. In particular, Benze, Ashwin, Sagnik, Samar, Amir and Farshad for always being supportive and ready to lend a hand. Without them, the past two years would've been a lot less enjoyable for me. Lastly, I want to thank those in my life outside Academia, including my relatives who looked after me here in Canada, my friends from undergrad, Matt, Frog and Bear (the cats, who made writing this thesis a little easier) and of course, my parents. They are the reason I am here, and their constant love, support and encouragement are why I was able to achieve this degree in the first place. I owe everything to them.

Also, an acknowledgement to McMaster University and McMaster Advanced Control Consortium for all the financial support.

Table of Contents

List of Figures

Declaration of Academic Achievement

The contribution of work in Chapter 2 is recognized as a novel solution to estimating parameters for high-fidelity models in Bioprocesses. It was published in CJCE in 2024 [1]. In this work, I am acknowledged as the primary author. My contribution includes Conceptualization, Implementation, Paper writing and revision.

Chapter 1

Introduction

1.1 Motivation

In the past decade, pharmaceutical products have gained popularity in the public health space due to recent health emergencies like COVID-19. Monoclonal antibodies are one such pharmaceutical product.

More recently, the Federal Drug Agency (FDA) has indicated the need to transition from traditional batch-based manufacturing to continuous manufacturing of these products [2]. This has opened up a variety of opportunities for developing continuous manufacturing processes for vaccine-related products. Sartorius, in particular, is currently in the process of developing its own continuous line for monoclonal antibody manufacturing. This recent advancement in Process Analytical Technology (PAT) for continuous pharmaceutical manufacturing has led to the use of high-fidelity models (both mechanistic and data-driven) for control of its updated manufacturing processes. One issue commonly faced with data-driven models is measurement noise.

This can cause uncertainty in model predictions, which can degrade controller performance. Therefore, the motivation of my thesis is to explore how modelling techniques can be used to make more reliable predictions, ultimately with the aim of having better control.

1.2 Background

An important step that occurs upstream in monoclonal antibody production is the live-cell culture of Chinese hamster ovarian cells (CHO). While traditionally done in batches, Sartorius has developed a perfusion (or continuous) based bioreactor for this purpose. This unit operation will be the test-bed on which new algorithms and techniques are trialled for this thesis[3].

Computer simulations, more specifically, models of the underlying processes are helpful. They aid in implementing new processes, or if existing processes need to be changed in any way. They can help in process monitoring and early-fault detection and are also necessary for implementing advanced process control techniques like model predictive control (MPC). First principles models (FPMs) are especially helpful in that they are explainable over a large range of operating conditions.

To that end, Sartorius has developed a detailed FPM of the process known as the Hybrid State-Space Model (HSSM). With all mechanistic models, for it to predict the specific trajectories that are needed, certain values or "parameters" need to be inputted into the model structure to calibrate it to the real-world process. The techniques used to find these values encompass the field of parameter estimation.

1.3 Using Bayesian Inference to estimate parameter distributions

In the field of bioprocesses, the problem of parameter estimation is a well-studied one. [4–6] provide parameter estimation methods for Bioprocesses, but without confidence intervals. Additionally, the majority of the work is done through a least-squares approach and results in giving point estimates of the desired parameters. The problem here is that with process data, there will invariably be some amount of noise caused by the measurement devices. This introduces uncertainty in the estimation itself. Therefore, we believed that it would be far more helpful to provide an estimate of this uncertainty along with the parameter, in the form of parameter distributions rather than point estimates.

There has been work done in this field. [7] and [8] both provide Bayesian-inspired methods to estimate parameters with confidence intervals. However, one key point is the models used in these studies. They are of a relatively lower complexity. As such, the computational expense of running such a model is low. Parameter estimation is in a sense, an effort in optimization. The model is recursively being run with different parameter configurations. If the model itself is of a higher complexity, the computational expense of running such a model will be high. This quickly accumulates and results in an intractable problem.

1.4 Other applications of Bayesian Inference for parameter estimation problems

Parameter estimation methods can be broadly categorized into 3 methods: Least-squares, Maximum Likelihood Estimation (MLE) and Maximum A Priori (MAP) [9]. The least-squares method typically tries to find the configuration of parameters that yield the least error value between the model predictions and the given data. The key assumption made here is the distribution of the error, which is assumed to be Gaussian. The MLE method recognizes this shortcoming, and its modification is that the error term is calculated while considering the underlying error distribution in the process data. The error term is calculated from a likelihood function, which is itself derived from the knowledge of the noise distribution. The MAP wraps the MLE in a Bayesian framework, by providing the posterior rather than the likelihood value[9]. The difference here is the incorporation of the prior and evidence terms. The prior function in Bayes' Theorem represents both the background knowledge and any information that can be obtained from historical data of the underlying process. The evidence is a predictor for the overall model performance considering the given data. While this isn't useful in typical parameter estimation problems where only one model exists, this can be a very useful metric when choosing between models.

1.5 Thesis Outline

The rest of the thesis will proceed as follows: Chapter 2 presents a proof of concept for the estimation of parameters in a bioprocess while providing reliable confidence intervals. Specifically, Bayesian inference is used to estimate the uncertainty in the prediction of a parameter due to the presence of measurement noise in the process. The resultant joint probability distribution is utilized to infer the confidence interval of the resultant estimates. This method is numerically applied using a technique known as Nested Sampling. This algorithm iteratively samples parameters from a pre-determined range of values to compare model predictions and obtain a probability density function. One challenge typically associated with this algorithm is in the determination of the prediction error, especially when a high-fidelity dynamic model is being utilized. For the motivating example in the present manuscript, where a high-fidelity simulated Bioprocess is being considered, the use of the high-fidelity model provided by Sartorius AG as part of the estimation algorithm poses computational challenges. To overcome this challenge, a universal approximator such as a parameterized neural network is used. This neural network is designed to simulate the results of the first principles model (while also capturing the dependence of the model parameters on the output), and once trained can provide near instantaneous results making Nested Sampling computationally tractable for this application. Simulation results demonstrate the feasibility and capability of the proposed approach.

Chapter 3 addresses the problem of having limited data for parameter estimation and offers a solution if historical data of the same/similar process is available. Specifically, the prior term used in Bayesian inference is used to inform the search space of the parameter estimates. The prior, in the form of a distribution function, is used to bias how the algorithm searches for parameter configurations. The Nested Sampling algorithm iteratively samples parameters to compare model predictions and obtain a probability density function. Conventionally, the algorithm samples uniformly from a pre-determined acceptable range of parameter values. However, if the data obtained is limited, there may not be enough information contained to make a reliable estimate. Therefore, one solution would be to make use of previous parameter distributions from older, but similarly operating processes. For the motivating example in the present manuscript, a high-fidelity model of a live-cell culture — run in a continuous upstream bioreactor — is provided by Sartorius AG. As a new cell line is introduced, there is a shortage of available process data. As such, the method outlined above uses existing parameter distributions from older cell lines to inform and bias the parameter estimation results for the newer cell line. This chapter also includes a section on the evidence, a term that is conventionally neglected for parameter estimation problems. However, the evidence can be important in model selection problems. Therefore, the final section of this chapter is related to applying and discussing the potential use cases for this term. Specifically, the evidence is calculated for a series of models with varying complexity to test if it can be a predictor for model performance.

Chapter 2

Leveraging Neural Networks to estimate Parameters with Confidence Intervals

2.1 Introduction

Pharmaceutical products have been increasing in popularity over the past few years with a focus on public health and innovation. Monoclonal antibodies are one such product developed by Sartorius. Bioreactors that produce cell cultures are important in the production of these monoclonal antibodies. More recently, process control through Model Predictive Control (MPC) has gained in popularity due to the added flexibility and reliability given during process operation. This has motivated the need to develop models that can in turn be used for process control [3].

First principles models are often desirable as they offer an understanding of the underlying process, and stand to be valid over a larger range of operating conditions.

In the context of bio-processes, first principles models (FPM) typically use a system of differential equations. However, to calibrate a model to a process operating under a specific condition, certain values need to be specified in the model. These values are known as parameters, and they can be inferred (or estimated) from experimental data on the process. The different techniques used to do so encompass the field of parameter estimation [10].

More recently, hybrid models have gained recognition, due to their ability to incorporate data-driven modelling techniques with well-understood mechanistic principles. This has allowed for novel estimation techniques like [11, 12] which describe a method to estimate parameters spatiotemporally in reaction-diffusion problems using a combination of machine learning techniques and partial differential equations.

In the field of process engineering, parameter estimation is a well-studied problem, and several studies have been published in this field. One example [13] details a parameter estimation algorithm for nonlinear chemical and biological processes with small datasets. It used Markov chain Monte Carlo (MCMC) methods to approximate posterior distributions for the desired parameters. Another example [14] is where a variation to the Lasso method (called the stable lasso method) is developed to estimate parameters that yield the most stable model. This method was applied for the dynamical modelling of an industrial boiler. Yet another [15] details a method used for modelling batch-to-batch process model parameter variation. This was shown to work within an MPC scheme.

Broadly speaking, there are three types of parameter estimation methods: Least-Squares, Maximum Likelihood Estimation (MLE) and Maximum A Posterior (MAP) methods. Least-squares methods rely on minimizing the error term between the predicted and actual data. The key assumption made here is that the prediction error that is determined from this method is normally distributed. MLE methods have an advantage over the former method, in that parameters can be successfully estimated even when the error is not expected to be distributed normally. This is because the objective of this method is to maximize a likelihood function, and is not restricted to one particular assumption on the error distribution. MAP methods take the concept behind MLE and wrap it in a Bayesian framework. The advantage here is that the prior or background knowledge of the process is captured and used to inform the parameter estimation [9].

There has been considerable work done in applying Bayesian approaches to process engineering problems. One example [16] uses a Bayesian approach to model identification for use in a computational fluid dynamic (CFD) model of a Steam Methane Reformer. Another example [17] applies Bayesian Inference to estimate posterior probabilities for fault detection and process monitoring of multimodal Gaussian process data. Looking at a different application of Bayes theorem, one article [18] details a method to estimate parameters online for process control and monitoring using Bayesian inference. Similarly, another one [19] makes use of the variational Bayesian framework to infer dynamic latent variable (LV) models that are characterized by abrupt variations in the process data. Yet another example [20], describes a novel online transfer learning technique to extract cross-domain features using Bayesian inference. Finally, an application of Bayesian Analysis is used to predict probabilities of risks involved with corrosion-induced incidents of liquid transmission pipelines [21]. This paper makes use of Artificial Neural Networks (ANNs) as well, in order to predict current conditions of the pipelines, which are combined with historical data for use in Bayesian analysis.

Results in the field of bioprocesses, [4–6] provide methods for parameter estimation, but without focusing on the determination of confidence intervals. In one result[22], a method for determining the confidence intervals when estimating parameters is presented, but this is done using conventional statistical methods. While this is a practical method for the mentioned problem in that manuscript, the approach is limited by the assumption of normality seen in most least-squares problems as well as the absence of prior information to influence the parameter estimation problem.

Finally, methods have been developed [7] and [8] to estimate parameters with confidence intervals using Bayesian approaches (albeit using techniques other than Nested Sampling). However, the key difference between this (as well as the previously mentioned work) is that the FPM model structures used were of relatively low complexity. As such, the computational expense involved in recursively running the model is negligent and they can feasibly estimate parameters. In certain cases, as with the current case, a first principles model may have been developed accounting for the known process complexities, but one that results in a fairly complex model. This process represents one example of a complex process of industrial interest. Sartorius has developed a detailed first principles model of this known as the "HSSM simulator". It includes a system of 10 Ordinary Differential Equations (ODEs) to describe trajectories of various cell and metabolite-related properties and quantities. Based on experimental data, this model has been shown to capture the inherent complexity of the process [3].

The parameter estimation problem for this kind of model is too computationally challenging using methods such as an MCMC approach [7, 8]. This is because these methods rely on recursively running the given model to determine a likelihood score. If the model itself is complex, then running the algorithm could result in it being too computationally expensive to generate a result in a reasonable timeframe.

One solution would be to make use of a surrogate model. A surrogate model is a simpler but still useful, possibly data-driven model for specific purposes such as the estimation of parameters, with Neural Networks being a promising tool for surrogate modelling. There have been other results [23, 24] that make use of Neural Networks for surrogate modelling. However, in these cases, the surrogate models are trained as if using a conventional data set. Thus, expectedly, care was taken to prevent the overfitting of these models to the available data. The present manuscript recognizes that the FPM is uniquely suited to provide practically an 'unlimited' amount of data, or at least as much data as needed not to be concerned about the problem of overfitting. Therefore, it can be argued that if the FPM is able to train the NN on every given scenario, and if the main task of the NN is to replicate the FPM, then the issue of overfitting has to be handled differently.

Based on the above considerations, this manuscript presents a Neural network-based surrogate modelling and parameter estimation method while addressing the issue of reliability due to the uncertain nature of the measurement data. Specifically, a Bayesian-based algorithm is used on a simulation test-bed. The algorithm is used to estimate the value of two key parameters (the maximum growth rate and primary death rate) of a detailed first principles model using limited (simulated) batch data with measurement noise. In order to deal with the issue of computational expense, a surrogate model in the form of a NN is used in place of a detailed first principles model. Finally, a method to test the reliability of the estimates from the NS algorithm is detailed. The paper is arranged as follows: Section 2.2 describes a bioreactor process, and gives a primer on Bayesian Inference and the Nested Sampling algorithm. Section 2.3 presents the specific implementation of the NS algorithm for this problem as well as presents the surrogate modelling solution including model structure and NN training. The results from an application of the proposed method are detailed in Section 2.4. Finally, concluding remarks can be found in Section 2.5.

It should be noted that Nested Sampling does not necessarily add any advantages over other MCMC-based methods (especially for parameter estimation problems) and as such is not part of the novelty of this work. It was chosen among other alternatives due to its ease of implementation and well-developed open-source Python-based packages like Dynesty. The key contribution of this manuscript is the use of a Bayesian-based parameter estimation algorithm along with a surrogate model to enable fast computations.

2.2 Preliminaries

A brief overview of the process and its FPM model is presented first. This is followed by a primer on Bayesian inference, the Nested Sampling algorithm and the Python package “Dynesty” that is used to implement it.

2.2.1 Process Background and Simulator

The bioreactor under consideration (henceforth referred to as the Bioreactor) presented in this manuscript is used for the production of monoclonal antibodies, an important step in the manufacturing of vaccines as well as other bio-based pharmaceutical products [25]. An important feature of this process is that it is run in perfusion mode. Perfusion refers to the continuous removal of bleed and harvest streams. This is highly desired as it leads to a greater harvest when compared to other conventional batch and fed-batch processes, as well as a lower cost-of-production [25].

This process consists of a live cell culture in an enclosed environment. After an initial 3-day period where the bioreactor is run in batch mode (known as the growth phase), the process switches to a perfusion mode for the next 30 days [3]. A simple schematic of this process is shown below (Figure 2.1).

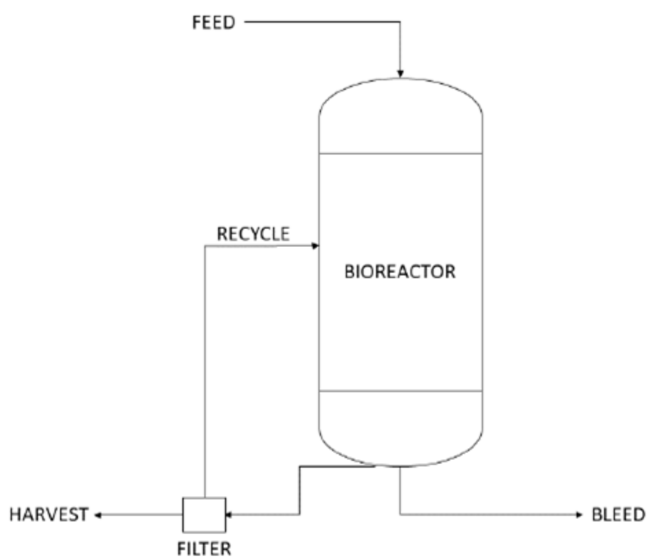


Figure 2.1: Schematic of a Bioreactor

As mentioned above, the Bioreactor is run continuously (also known as to be in perfusion mode). The Bioreactor has a feed input designed to maintain Glucose concentration, an essential component in the growth of the cell culture. The recycle stream is used to recycle live cells back into the process as the live retention filter does not allow for live cells into the harvest (also known as the titer). There are many performance considerations, but broadly the objective of this process is to maximize the titer over the perfusion phase. The viable cell density (VCD), which is directly related to the titer, is controlled by a PI controller the bleed rate as the manipulated input [3].

The key challenges are determining the values of the parameters, especially for other cell lines, that may have the same first principles governing equations, but different values for parameters such as growth rate and death rate. For this manuscript, the HSSM simulator is used as a test bed reactor, both to generate data and also as the first principles model structure with parameters that need to be estimated. The Nested Sampling algorithm will be used to estimate "unknown" parameters from data generated by this test bed. More details about this are given in the next section.

This section describes at a high level, the HSSM model, developed by Sartorius. While the explicit equations used in this model are not available for disclosure, more details about the modelling of this bioprocess can be found in [26].

2.2.2 Bayesian Statistics and Inference

The data that is received from Bio processes such as these contain some amount of noise due to measurement error. Therefore, it is appropriate to estimate both the best-fit parameter value according to the data as well as the uncertainty in the estimation of said parameters (resulting from the measurement noise). In this manuscript, the uncertainty in the parameters (informed through the use of confidence intervals) is determined using Bayesian Probability theory. In probability theory, there are two main approaches – Frequentist and Bayesian probability theory. The more popular frequentist theory attempts to estimate probability based on the long-run frequency of events. This theory is useful for calculating the probability of an outcome occurring given parameters and hypothesis [27]. However, a more important problem is the inference of a cause/multiple causes given some effect, also known as inductive logic. Bayesian Inference was developed as a solution to the above problem. The derivation for Bayes' Theorem is summarized below: [28]

The basic conditional probability can be written as:

$$p(B \mid A) = \frac{p(B \cap A)}{p(A)} \tag{2.2.1}$$

where the probability of Event B occurring is conditional on Event A. Similarly, we can write:

$$p(A \mid B) = \frac{p(A \cap B)}{p(B)} \tag{2.2.2}$$

Since $p(B \cap A) = p(A \cap B)$, this becomes:

$$p(A \mid B) = \frac{p(B \mid A) \cdot p(A)}{p(B)} \tag{2.2.3}$$

The above is the Bayes' rule, created by Reverend Thomas Bayes in 1763 [28]. When dealing with data and model parameters, we substitute A with the data set D, and the model parameters with θ as below:

$$p(\theta \mid D) = \frac{p(D \mid \theta) \cdot p(\theta)}{p(D)} \tag{2.2.4}$$

There are three main components to the Bayes' Theorem (that are useful in the context of parameter estimation). The prior, the likelihood function and the posterior distribution.

The Prior is the $p(\theta)$ element in Bayes Theorem. It is defined as the probability that a given hypothesis is true based on prior beliefs. In the context of the present manuscript, this refers to the background knowledge that is available on the desired parameter. Thus, it defines a PDF for a range of parameter values based on prior knowledge. For the present manuscript, other than a general understanding of the ranges where the growth and death rate parameters should lie, not much else is known. Therefore, the prior function is defined as a uniform PDF within some defined range. [28]. The Likelihood is the $p(D \mid \theta)$ element in Bayes Theorem. It is defined as the conditional probability of a sample matching some observed data, given that the sample is generated using a given hypothesis. In the context of the present manuscript, it is assumed to be the conditional probability that the sample trajectory created from the HSSM simulator using a chosen set of parameters matches some observed data [28]. The Posterior is the $p(\theta \mid D)$ element in Bayes Theorem. It is defined as the conditional probability of the Hypothesis being true, given that the sample (generated from said hypothesis) is a match for the observed data. In the context of the present manuscript, the posterior is assumed to be the probability that a given parameter is the "true" parameter if said value generates a sample that matches a given set of observations. This PDF is the function that this project is looking to obtain [28].

Based on Equation 2.2.4, it is evident that the posterior is calculated by weighing the probability score of the likelihood against the prior probability. This is important in cases where the likelihood score is relatively high, but based on previous observations/experiments, this value for the parameter is rarely seen, giving it a low prior probability score. In the context of the present manuscript, it is possible to see a similar scenario where for a given run, a parameter value has a high likelihood score, but based on previous simulated data, this value (or range of values) for this parameter is rarely seen. Since the prior and likelihood scores are multiplied with each (see Eq. 2.2.4), the posterior is always balanced by these two terms. As such, a high posterior score is obtained when both the likelihood and prior terms have high values, and vice-versa. Additionally, having a high likelihood value and low prior value (or the opposite) can give the posterior a moderate value rather than a relatively high or low score.

The denominator in Bayes theorem is known as the "Evidence" (also referred to as "Z"). It is defined as the probability that the observed data is "true" for all possible values of the parameters. It is also sometimes referred to as "marginal likelihood" or "prior predictive", although there is much disagreement in the technical definition for this term [29]. Conventionally, computing this term requires added effort, involving a "sequence of intermediate distributions that bridge between prior and posterior" [29]. This added effort, coupled with the lack of a standard terminology led to the evidence becoming an optional term. This is seen with Markov Chain Monte Carlo (MCMC) methods that typically determine a normalized posterior by ignoring the evidence [29].

In Bayesian Inference, the main goal is to calculate the posterior distribution. This is the probability distribution that determines which value for a given parameter was most probable to cause this particular measurement. The advantage is that it provides both the optimal point for the parameter (maximum a posteriori or MAP) as well as the distribution curve for all values within the parameter space. This makes it possible to understand which range of values are most probable. When dealing with rudimentary models, it is possible to derive simple expressions when computing the posterior distribution. For more complex models, however, computer-based numerical methods are needed [30]. One such technique is described next. As mentioned above, the term 'Z' refers to the evidence. It is defined as the probability of the observed data being 'true' for all parameter values of the given model. In the context of parameter estimation, since we are working with only one model (either the FPM or the surrogate alternative), this term is of little use. However, when comparing between competing models (as in a model selection problem), this term is important. This term is calculated by integrating the likelihood function over all prior ranges. Therefore, a common problem is its intractability when using more than very few parameters, as the integral becomes difficult to calculate over that many dimensions.

2.2.3 Nested Sampling

The goal of Nested Sampling is to estimate the evidence. As mentioned previously, the evidence "Z" represents the probability of the data integrated over all values of the parameters. While this term is usually neglected [31], calculating Z allows us to evaluate the posterior of a given model accurately, and since Z is constant (as the PDF for the observed data remains constant), different model assumptions can be evaluated without re-calculating the evidence. [32].

Conventionally, Riemann-style integration methods are used, in which the parameter space is divided into small cubes and a sum is performed over all of them.

However, this technique becomes infeasible in multiple dimensions as the cost of performing these summations becomes exponential, suffering the curse of dimensionality [29]. Other methods like Markov chain Monte Carlo (MCMC) have been designed to estimate the posterior. However, MCMC gives only a few samples that represent the posterior and fails to determine the evidence. As discussed before, the evidence is a useful value to have, and Nested Sampling makes the estimation of this value the primary target [32].

The Nested Sampling Algorithm works by drawing a collection of N live points from a sample space. The likelihood values for each of these points are calculated. The live point with the lowest likelihood score is then dropped from the sample, and a replacement point is drawn from the prior subject to the constraint ($L > L*$), where "L" refers to the likelihood score obtained from sampling a parameter θ, and "L*" refers to the likelihood score of point that was dropped in the previous iteration (this is also the lowest likelihood score from the collection of live points). Each time a live point is dropped, the volume of the sample space is compressed by a factor of t, known as the shrinkage factor [29]. Therefore, as more points are dropped and replacement points are found, the successive compressions lead to successively smaller inferred volumes "X" [29]. The specific procedure for Nested Sampling is as follows [32]:

Start with N points $\theta_1, \ldots, \theta_n$ from prior.

Initialise $Z = 0, X_0 = 1$

Repeat for $i = 1, 2, \ldots, j$

record the lowest of the current likelihood values as L_i

set $X_i = exp(-i/N)$

set $w_i = X_{i-1} - X_i$

increment Z by $L_i w_i$

Replace the point of lowest likelihood with a new one drawn from within $L(\theta) > L_i$, in proportion to the prior $\pi(\theta)$.

Increment Z by $N^{-1}(L(\theta_1) + +L(\theta_N))X_j$

where θ refers to the parameter being estimated, Z refers to the evidence, X_i is the updated prior volume at iteration i, w_i are calculated weights and L_i is the likelihood function.

2.2.4 Dynesty

In order to implement Nested Sampling for this project, a package known as Dynesty was used. Dynesty is a public, open-source, python package that implements a version of nested sampling to estimate Bayesian posteriors and evidence [33]. As mentioned previously, Nested Sampling provides many advantages over conventional posterior estimation techniques like MCMC. However, traditional nested sampling has three main drawbacks: A prior transform is usually required, the runtime is dependent on the size of the prior space and the rate of posterior integration is always constant. While the first two limitations are inherent to Nested Sampling, the third one is not. This is because the rate of posterior integration is only constant when we assume that the number of live points remains constant during the estimation run. To that end, the conventional algorithm was modified [34] by letting the number of live points vary during the run, thereby solving the third issue. This is known as Dynamic Nested

Sampling and is one of the defining features of Dynesty. With this, Dynesty has implemented an algorithm that allocates the number of live points for each iteration [33].

2.3 Present Work

This section describes the representative data, the nested sampling setup, the Neural Network surrogate model development, and the parameter estimation results.

2.3.1 Generating Simulation Data

The HSSM simulator was used to generate representative data, of the kind that would typically be available for the parameter estimation problem. Thus, a set of 3 experiments or runs were generated using the first principles model. As this data is obtained artificially and not from experiments, certain features were introduced to make this data as applicable as possible to an actual scenario. First, the growth and death rates were held constant between the three runs. This was to ensure that while each run was unique, the inherent properties or "truths" among them remained constant. Each run had a different starting point. A starting point for the initial values of VCD and Glucose was determined using the 12 AMBR 250 experiments that were used to develop the HSSM simulator. To this, a random value was added. The trajectory that each run took was different. This was done by changing the VCD setpoint randomly by up to 5% every 3 days. At each time step, the value of VCD that is measured and sent to the PI controller has "measurement" noise added to it. This is done by adding a random Gaussian value with a standard deviation of 2.

2.3.2 Setting up Nested Sampling

The experimental data is then passed into the Nested Sampling algorithm. As mentioned above, the specific package used to apply Nested Sampling is called Dynesty. Along with certain Dynesty Parameters, two main functions need to be defined in order to run this method.

The prior transform function is used to define the range of values that can be used when estimating a parameter. The algorithm will be limited to searching only within this space. This function can also define a probability density function (PDF) for the range of priors. However, for this manuscript, we are using uniform priors, meaning that the range of parameter values defined has a uniform probability. The Likelihood function as mentioned in previous sections is a PDF that determines how likely a sample is said to match a set of data, given some parameter that was used to generate the sample. For our purpose, the growth and death rate would serve as the parameters for this function. The sample then, would be the trajectory of VCD generated using the parameters found by Nested Sampling, and this would be compared to the experimentally generated data.

This function is derived using the PDF used to obtain a particular sample. Since our samples contain Gaussian measurement noise, the PDF is defined as such:

$$PDF = (2\pi\sigma^2)^{-1/2} exp\left(\frac{-1}{2}\frac{(x-\mu)^2}{\sigma^2}\right) \tag{2.3.1}$$

where σ refers to the standard deviation, μ refers to the mean (in the present instance the experimental data) and x refers to the results from the FPM and the errors is assumed to be drawn from the Normal population. Since the joint probability of obtaining each element within the sample (assuming each error element is an independent measurement) is the product of each PDF, the equation for the likelihood function (L) is:

$$L = \prod_{j=1}^{n} \frac{1}{\sqrt{2\pi}\sigma^2} exp\left(\frac{-1}{2}\left[\frac{(x_j - \mu)}{\sigma}\right]^2\right) \tag{2.3.2}$$

Then converting to a log-likelihood function (l) turns the product into a sum, given by:

$$l = \frac{-n}{2}\ln 2\pi - \frac{n}{2}\ln \sigma^2 - \frac{1}{2\sigma^2}\sum_{j=1}^{n}(x_j - \mu)^2 \tag{2.3.3}$$

The above equation is used to calculate the log-likelihood function in Nested Sampling.

Nested Sampling calculates the likelihood in a way that uses MCMC methods. However, rather than sampling from the entire likelihood directly (which can be a challenging and computationally expensive task), NS uses a constraint $L(x) > L*$ in order to "bound" the likelihood search space. From here, this space is iteratively shrunk by increasing the threshold of L*, essentially looking for samples that have higher likelihoods within our earlier defined bounds. This method can be likened to solving a simple problem multiple times, as compared to typical MCMC methods which solve a hard problem once [35].

The objective of the likelihood function is to calculate the probability of a sample (generated from a model through a parameter value θ) matching the given data. The function itself differs from other least-squares methods in that it does not assume normality by merely calculating the error term between the predicted and actual data, but seeks to minimize an objective function. This objective function is derived from knowledge of the distribution of the data. Therefore if a non-gaussian distribution of errors is expected - and knowledge of the underlying distribution is known - the likelihood function can estimate parameters for this.

2.3.3 Limitations with using Nested Sampling

As mentioned in previous sections, Nested Sampling is an iterative application of Bayesian Inference. Therefore, for the algorithm to converge, it must repeatedly sample points from the prior space, and in a direct application would use the first principles model, in this case, the HSSM simulator to generate samples for the log-likelihood function. This step is particularly computationally expensive, as this involves solving a system of 10 ODEs and Nested Sampling typically converges after 50,000 such function calls based on our results.

Therefore, in order to reduce the computational load, a Neural Network (NN) is used in combination with the Nested Sampling algorithm. Specifically, a NN trained to replicate the HSSM simulator is used as a surrogate model. The key benefit to using this solution is that since the NN model is only meant to replicate the simulator, there is essentially an infinite amount of data that can be used to train the NN. This means that the NN can be designed to be "prepared" for every scenario during the training phase.

2.3.4 Neural Network: Inputs and Outputs Selection

As the purpose of the NN model is to replicate the HSSM simulator, the appropriate inputs and outputs were determined by looking at the first principles model itself. The outputs of the NN, which are the same as the outputs to the HSSM simulator are the trajectories of certain variables that occur within the bioprocess. Specifically, the model will be used to predict the trajectories of VCD and Glucose.

The inputs to the model were determined by looking at the first principles model. Based on the inputs used to generate a result from the HSSM simulator, certain variables/values were collected as potential inputs into the model. These included the Initial Conditions for VCD and Glucose and the trajectory given to the VCD setpoint in the PI controller. In addition, since the neural network is desired to replicate the ability of the simulator to compute trajectories of the variables for various values of the parameters (umax and death rate), these are also used as 'inputs' in developing the NN.

Thus, there are a total of 35 potential inputs into the NN model. Note that the present application recognizes that the intent of the NN is to fit (as much as possible) the dynamics of the first principles models, and thus is set up to give the NN the best chance to do this. Giving input data that is correlated would not benefit the NN design in any fashion, and it stands to reason to ensure that the input data is uncorrelated. To achieve this, the chosen input data was passed through a PCA filter. As the figure below (2.2) shows, the dataset can explain itself in 12 scores, implying that there are 23 inputs that are correlated, and therefore should not be defined as the inputs to the NN.

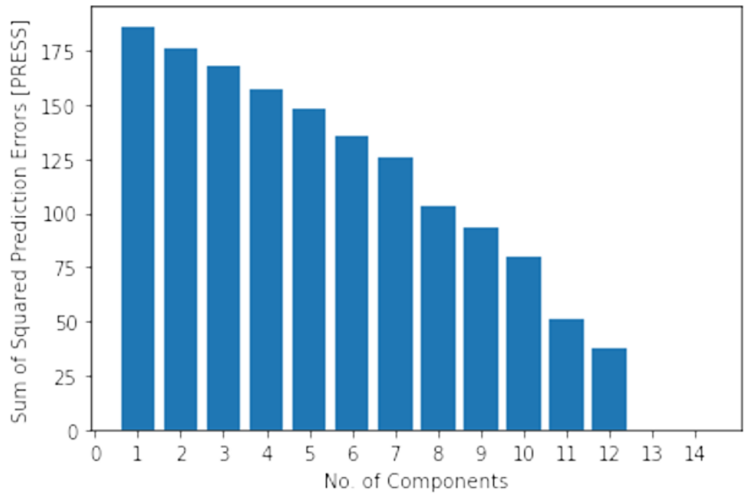


Figure 2.2: PCA Analysis of Input data to Neural Network

The PCA analysis enables realizing what in retrospect seems obvious: while the VCD trajectory does change every 6 days, it does not change between these intervals. Therefore, the inputs were redefined to just include the first timestep at which the VCD setpoint changes, rather than the entire trajectory. This shortened the number of inputs needed to 12. Recalculating PCA scores reveals that 12 scores are needed for this new set of inputs, verifying that each input is independent. Finally, as the first VCD setpoint is common for all runs, this was removed as well and the input space was further shortened to 11 inputs.

2.3.5 Neural Network Structure and model training

In order to determine the best structure for the NN model, a series of trials were set up in which different hyperparameters were adjusted. From this, the structure shown in Figure 2.3 below was obtained.

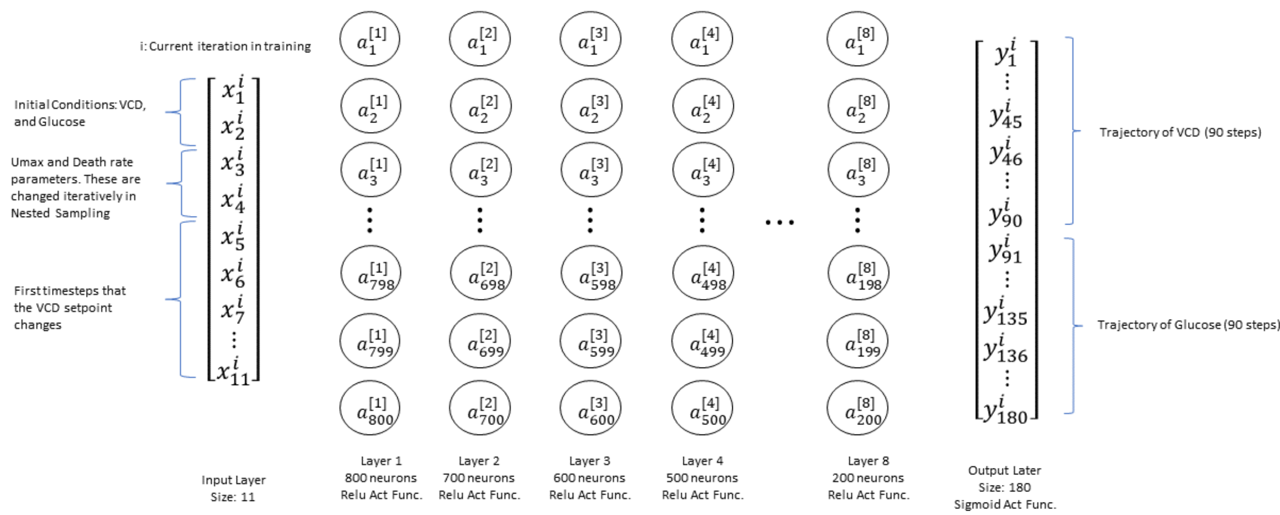


Figure 2.3: Proposed structure for NN

As illustrated above, the model has 8 hidden layers, and a progressively decreasing number of nodes as the layers progress towards the output. The number of nodes decreases from 800 nodes in the first layer to 200 nodes by the 8th layer. The activation functions were chosen to be ReLu in all of the hidden layers, and sigmoid in the output layer.

As mentioned in previous sections, the goal of this NN is to replicate the FPM. Therefore, the training data used is generated from the HSSM model itself. A collection of 50,000 runs was generated with varying set-point trajectories, parameter values and initial conditions to appropriately excite the data for this purpose. Additionally, a further 5000 runs were generated for testing purposes. It is important to stress that this task was fairly trivial to accomplish given that no experiments were needed to 'collect' this dataset.

A variety of training methods were deployed when developing the NN. From these, a variation of the mini-batch method was selected. The training algorithm is described below:

1. While loop keeps going till the validation score does not change in the past 500 epochs
   (a) Randomly samples 1000 points from full Training Dataset
   (b) Takes sampled data in batches of 32
      i. Forward Calc
      ii. Loss Calculation
      iii. Back Propagation
      iv. Optimizer to update weights (done after every batch)
   (c) Calls rate scheduler function to update learning rate
   (d) Evaluates testing data based on current weights to get a score
   (e) Checks if the score got better in the past 500 epochs

The main difference between this method and the mini-batch method is that for each epoch, the entire training dataset is not used. Rather, a subset (1000 batches) is randomly chosen from the entire dataset at the beginning of each epoch. From there, the standard mini-batch method using 32 batches at a time is implemented. While this method does not yield significant improvements over the mini-batch algorithm, it does reduce the training time significantly.

2.4 Results

This section first presents the NN surrogate modelling results, followed by the parameter estimation results.

2.4.1 Neural Network Surrogate Model

Based on the Neural Network structure and training method described in the methodology section, a model was obtained. The fact that the NN is being used to build a surrogate model for a first principles model (as opposed to training from an existing data set) has significant implications on the design of the NN. It changes the entire approach of NN design, in the sense that overfitting is not an issue, and that 'exact' replication is a possibly feasible, and desirable objective. Traditionally, overfitting concerns arise due to the NN model having to predict scenarios for which it was not trained, leading to errors in extrapolation, and NN possibly overfitting the noise associated with experimental measurements. Clearly, noise is not a concern with simulated data, and because the data generation and training exercise is offline, sufficient data could be generated to practically cover the entire range of expected utilization of the surrogate model. This approach was followed in the present manuscript so that the trained NN would be able to replicate any result seen from the FPM to a high accuracy, and in turn, enabling the determination of the parameters of the first principles model.

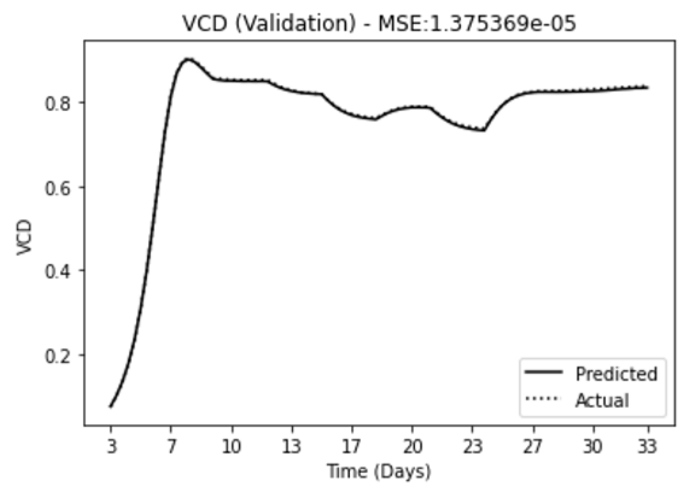


Figure 2.4: Sample run from the testing dataset of the VCD trajectory

Once trained, the model was evaluated using the testing dataset. The mean-squared error (MSE) was used to determine the model's performance, essentially representing the average error between the predicted and generated data. The MSE was determined to be 1.375×10^{-5} for VCD (See Figure 2.4). This error value was determined to be negligible and hence this model was chosen to sufficiently represent the HSSM simulator.

2.4.2 Nested Sampling Results

Recall that there are two main objectives of the present manuscript. The first, and main objective, is to detail a method to estimate parameters using complex models. The second objective is to deal with the issue of high computational expense through the use of surrogate modelling.

As mentioned in previous sections, a limited dataset generated from the HSSM simulator is treated to be the experimental data for the purpose of this manuscript. The parameters used to generate this data are not given to the Nested Sampling algorithm. These values, considered to be the "true" values are 0.81 and 0.06 respectively. After running the algorithm (as mentioned in the methodology section), a joint probability contour plot is generated.

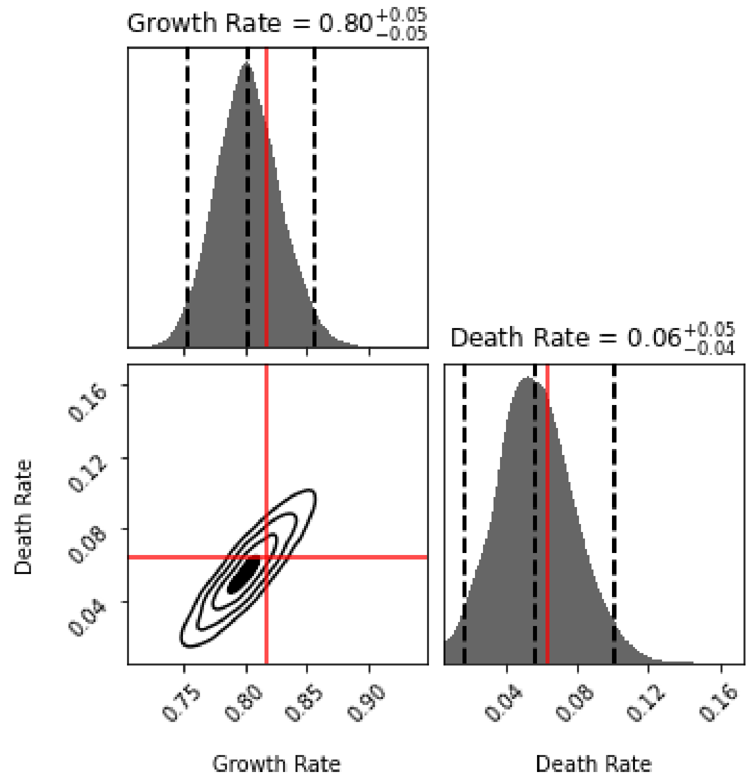


Figure 2.5: Joint Probability Distribution for the estimated parameters

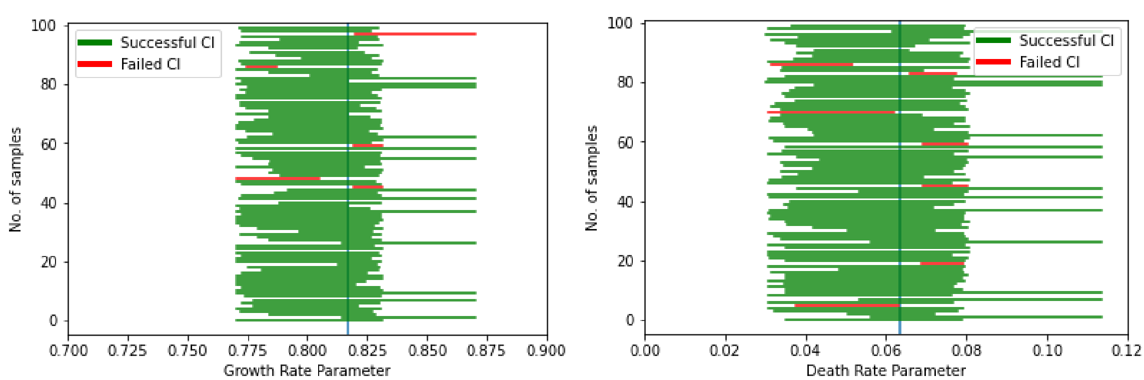
Figure 2.5 represents the joint posterior distribution of the growth rate and death rate for the dataset shown to the algorithm. The darkened regions show areas of high joint probability that the parameters are the right fit for the dataset. The red line represents the "true" value of the respective parameters used during the generation of the "experimental" data from the HSSM simulator.

After the Nested Sampling algorithm is converged, Dynesty produces data on measurements conducted during the run. In this, the "coordinates" of the parameters chosen for a particular iteration along with their corresponding "importance weight" (a metric that is directly related to the likelihood function) is shown. It is helpful to visualize these results through the use of scatter and contour plots. Plots like Figure 2.5 are good at showing the spread of data due to varying levels of measurement noise and possible correlations formed between the parameters. However, a major objective for this manuscript is the estimation of parameters, specifically with confidence intervals. Therefore, a quantitative method to analyze these results is required.

The effect of using a surrogate model within the NS algorithm is significant. After testing, it was found that using the HSSM model directly with the parameter estimation algorithm resulted in it converging in **3 hours and 16 minutes**. When using the NN however, this convergence time is reduced to **3 minutes and 5 seconds**.

The manuscript demonstrates the use of surrogate modelling for rapid estimation of parameters. This is important for several areas, a key one being process control. The present work would allow for the estimated parameters to be used in the deployment of models online. In cases where the environment and hence the parameters within the bioreactor are changing, this can be especially advantageous. The rapid estimation of parameters can also be beneficial in speeding the calculation of results when designing new scenarios for the bioprocess, allowing for the rapid design of experiments.

A Confidence interval is a range of parameter values that are shown to be reasonably consistent with the data observed. These intervals are used alongside values called confidence levels. This is the long-run probability that the chosen bounds capture the "true" value of the population parameter [36]. Therefore, adapting this for the present work, if one were to require an estimate for a set of parameters with a specified confidence interval, say 90%, then this method should be able to provide these estimates with bounds such that if this trial was repeated n times, the “true” value of this parameter must be included within these bounds 90% of the time. It should also follow that higher confidence levels will lead to larger bounds and vice-versa. In order to calculate the parameter multiple times, the first choice would be running the Nested Sampling algorithm a desired number of times and determining the bounds for each run. However, this method is computationally expensive, and would not allow for the rapid estimation of parameters which is a major objective of this project. A solution to this is the use of resampling methods such as Bootstrapping. Bootstrapping is a statistical technique where samples are drawn with replacement from an original dataset, usually to estimate a population parameter. This is typically done in cases where only limited data is available, or it is too difficult to obtain more [37].



(a) Growth rate estimate

(b) Death rate estimate

Figure 2.6: Visual representation of resampling method. Shown here is 100 runs instead of the original 1000 for clarity.

Using the ”`resample_run`” function in Dynesty, the results from the initial NS run are resampled using this technique. The resampled datasets are considered to be new runs of the NS algorithm and are a much more efficient way to recalculate the parameter estimate bounds. These resampled datasets are then arranged by their importance weight and a 'cutoff' is implemented so as to select only the samples with the highest weight. It should be noted that this cutoff value is selected based on the desired confidence level of the bounds. From the selected samples, the highest and lowest values for each parameter are chosen as the parameter confidence intervals for that run. This is repeated for a certain number of times and the proportion of failed against successful bounds determines the confidence level. Using the NS run that Figure 2.5 was generated from, this method was applied to it. The results were resampled a 1000 times and after rearranging these datasets by importance weight, a cut-off of 10 was applied to it. This meant that only the 10 highest weighted samples were taken from each dataset to calculate the parameter bounds. From this trial, it was found that 46 out of the 1000 bounds failed indicating a confidence level of 95.4%. Therefore, based on the generated 'experimental' data given, this algorithm estimates that the growth rate is between $[0.775, 0.871]$ for a confidence level of 95.4%

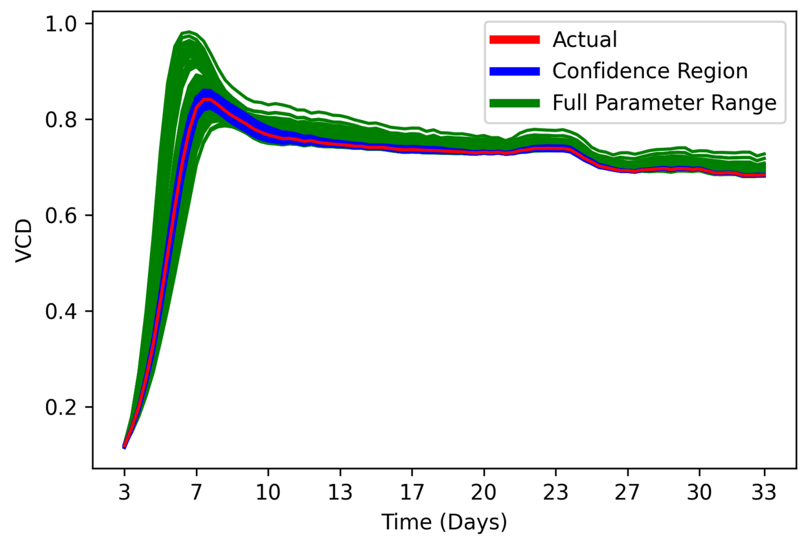


Figure 2.7: Visualizing the reliability of NS results by passing estimates into the surrogate model to predict trajectories of VCD

A significant and practical challenge in Model Predictive Control designs is dealing with system uncertainty. There are existing results [38] that use physics-informed NNs online to quantify uncertainty in the process data for use in stochastic MPC techniques. Another example, [39] developed a 'Safeness Index '-based controller that can be used to deal with stochastic non-linear systems with input constraints. If a description of the uncertainty is available, stochastic model predictive control (SMPC) can be deployed [40]. The present work presents a computationally tractable method to provide this knowledge in the form of parameter estimates along with information about their uncertainty in the form of confidence intervals.

The method shown above can be used to identify if the bounds obtained contain the "true" value of the parameter. When dealing with bioprocesses, this representation of the parameter estimates may be helpful as an alternative to offline analytical measurement tools. However, if the goal of the parameter estimation is to use these values in the development of complex FPM models, then a different way of analyzing these results might be preferred. For example, a better test would be to see the quality of the predictions that arise from using the estimated parameters as compared to using an 'incorrect' parameter value. Shown in Figure 2.7, the bounds estimated from the NS algorithm were used to predict trajectories for VCD. This was compared to the "experimental" (generated) data, as well as from sampling random values across the usable parameter ranges. By analyzing the variation in predictions, one can adjust the bounds to determine acceptable thresholds based on the model development problem at hand.

Note that due to the Gaussian nature of the distribution of noise in the data, the likelihood function used for this project is a function of the error term between the predicted and measured data. However, the likelihood function used is derived from the Gaussian PDF, meaning that this function factors in the standard deviation (as shown in eq. 2.3.3). This allows for the estimation of the spread of the data, which is not possible when using prediction error metrics like the mean squared error (MSE). The likelihood function allows for the estimation of parameters in cases where the measurement noise is distributed in a non-Gaussian fashion. Additionally, historical knowledge of the parameters could directly be used to bias the algorithm through the use of the prior function.

2.5 Conclusion

This manuscript provides a method to estimate parameters from noisy measurement data reliably through the use of confidence intervals, while also providing a solution to the issue of high computational expense from recursively evaluating complex models by introducing a surrogate model in the NS algorithm. For this, a NN was trained to replicate the HSSM model. Due to the function of this NN as a surrogate model, the training and testing methodology used deferred from conventional NNs. Finally, the NS algorithm was used along with the proposed surrogate model. These results were analyzed and resampled using a bootstrapping technique to provide confidence intervals for the estimates according to a desired confidence level.

Acknowledgments

Financial Support from McMaster Advanced Control Consortium and Sartorius Canada Inc. is gratefully acknowledged.

Chapter 3

Addressing practical concerns in Parameter Estimation problems

3.1 Introduction

So far, the previous chapter has focused on using the likelihood function in the parameter estimation problem. As mentioned in the previous sections, the posterior as calculated by Bayesian inference methods is a combination of three terms: the likelihood, the prior, and the evidence terms. This chapter seeks to understand and apply the other two terms to the work presented in Chapter 2.

The prior in Bayes theorem is defined as the probability that a given hypothesis is true based on prior beliefs [9]. It can be interpreted to be the information obtained from previous (or historical) runs. The information here is the knowledge of parameter distributions. As mentioned in the introduction, when a change is made to an existing process, the parameters inherent to the system need to be re-estimated. However, there may likely be a shortage of data, especially if the underlying process is expensive to run. One solution would be to adapt the information obtained from older, but similar processes to the present problem. The prior function, which decides how and where the algorithm should sample potential parameter values, can be used to bias current runs towards more likely regions (as determined from previous runs). Specifically, the posterior distributions from older runs can be used as the prior function for the new run. Based on the above considerations, this chapter presents a method to use historical data to inform current parameter estimation problems, especially when limited process data is available. Looking at similar studies done, the Bayesian prior has been used extensively throughout the field of process engineering [41–43]. However, there are relatively fewer studies on the prior function when looking at Bioprocesses specifically. Two recent studies stand out in particular[44, 45]. While these studies have successfully shown the effectiveness of the prior function, there are certain differences with the present work. [45] does make use of a normally distributed prior function, but assumes that these prior distributions are available to the user. Comparatively, the present work derives the prior function directly from historical data, hence providing an all-in-one solution requiring just the data and the first principles model. [44] is another important study done in this field. In this, the author is able to demonstrate a feasible method to obtain a prior distribution from the data. The difference here is that the prior function is partly derived from user-based knowledge about the process. This is not always feasible, especially if the parameters in question are not physically real values.

In the context of the present work, the same bioreactor is being considered as in the previous chapter. This time, a new cell line is introduced, with new parameter values. As this is a more recent implementation, there is a limited amount of data available. Therefore, this method attempts to adapt the parameter estimation results from older cell lines to the newly available limited dataset

3.2 Present Work

3.2.1 Generating Simulated Data

As in the previous chapter, the HSSM simulator is used to generate simulated data, of the kind that would be available during a typical run of the bioreactor. Similarly, to simulate the effect of measurement noise, a random value is added to each point in the predicted trajectories. These random numbers were generated from a normal distribution with a standard deviation of 2. For the present work, there will be data from 2 similar, but distinct process configurations. The first is a larger dataset, consisting of 9 runs of the bioreactor. This collection represents historical runs from an older cell line. The second is a limited dataset, consisting of just 1 run of the bioreactor. As this collection represents a newer cell line introduced to the bioreactor, it will have different (but similar) inherent properties, including the parameters that are to be estimated from the former dataset. Another change from the previous chapter is the number of input perturbations to the system. The data used before changed its VCD set point every 3 days, resulting in 9 input perturbations. To reflect more closely the real-world operation of such a bioreactor, the number of perturbations was reduced to 2.

3.2.2 Generating the Prior Function

The main difference in this method is how the prior function is defined for a parameter estimation run. As mentioned before, the prior function is typically used to define a range of acceptable values for the algorithm to sample within. For Chapter 2, this was defined as a uniform range, meaning all values within these bounds had an equal probability of being sampled by Nested Sampling. This can cause issues, if the data received is limited. The presence of noise can also be a hindrance to the algorithm, as with fewer data points, the chance that a "wrong" value might give a high likelihood score is greater. One solution is to "guide" the algorithm towards searching for points in likely regions. These likely regions are determined from previous experiments, where certain parameters may not have changed significantly.

Therefore, in order to come up with a prior function, one can use the posterior distribution from older runs. This is because the posterior describes not only the Maximum A. Posterior (The "most likely" parameter value), but also a distribution of likely values across the acceptable range of parameter values. It collects the information received from sampling throughout the range during a Nested Sampling run and condenses it into a distribution. One thing to note is that the posterior is discretized, meaning that we see a collection of points after the algorithm has converged. It is then our job to convert this mass of points into a usable distribution function for the prior term. As the synthetic noise and the likelihood function are both derived from a normal distribution, the posterior will typically also be arranged in that distribution (see fig. 3.1). By calculating the mean and standard deviation, we can model this into a usable function for the prior.

3.3 Results

3.3.1 Prior Results

To obtain the final results for the proposed method, a prior function must first be built. As mentioned in the previous section, this is done by estimating a posterior distribution over older results. Therefore, in order to demonstrate this, a simulated dataset was constructed. It contained 9 runs of the bioreactor, and all had the same parameter values for the growth (0.9) and death rate (0.1). This is to signify that these runs come from the same cell line. However, to induce some variation in the information contained, each run had a different set of initial conditions for its VCD and Glucose concentrations. The system was also perturbed (through its VCD set point) to different random values for each run. Based on running this data through the standard NS algorithm, the following posterior distribution plot is formed:

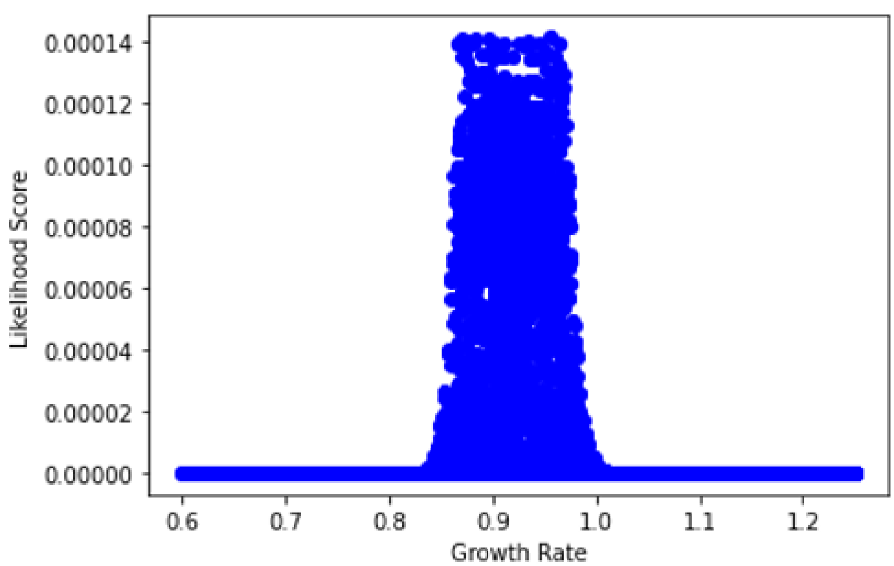
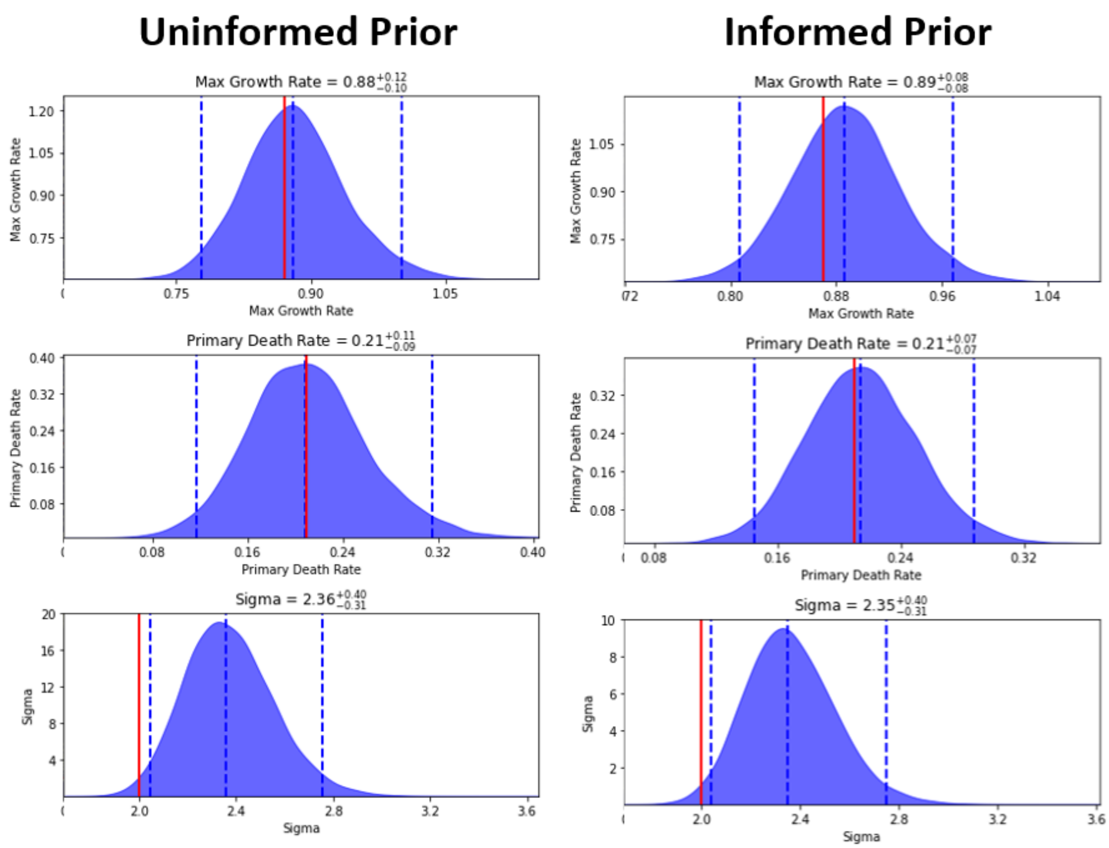


Figure 3.1: Prior Function, Mean = 0.930, Std Dev = 0.0896

Based on the simulated data given to the algorithm, the parameters are estimated relatively well, having captured the "true" value for growth and death rate within its distribution whilst also having smaller confidence bounds (as compared to a 3 or 1-batch run). This is to be expected, as more data and as such more information is being given to the algorithm, which results in more reliable predictions. This distribution can be estimated by calculating its mean and standard deviation. This is a relatively trivial calculation, but this means the distribution is assumed to be Gaussian. However, the discretized nature of this method allows us to see distributions that are non-Gaussian as well. A potential future study could look into estimating non-Gaussian prior distributions

3.3.2 Proposed Method — Results

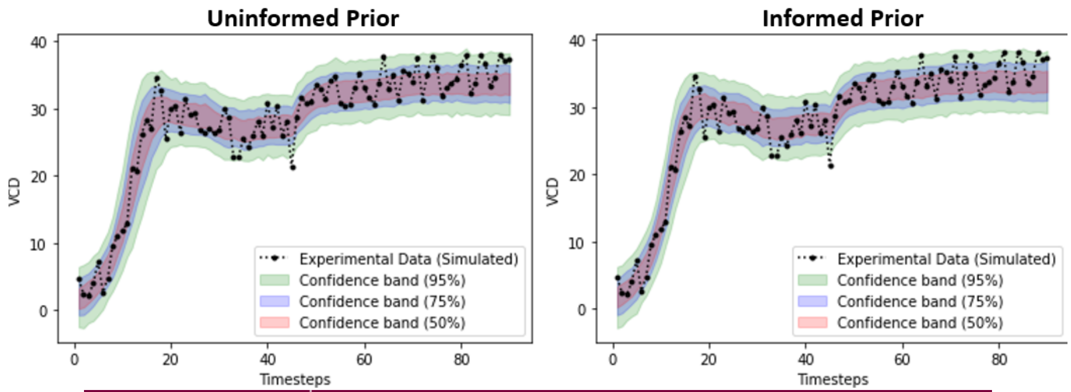
In order to demonstrate the effectiveness of this method, two scenarios were considered. The first considered a system that was perturbed less than default. Throughout the work done on the HSSM model, the simulated bioreactor was perturbed 9 times during its 27-day run (i.e. every 3 days). This number was reduced to just 2 perturbations per run. In fact, it was found in a study done on the same model that there existed a trade-off between how many times the system could be perturbed and the overall health of the cell culture [46]. As mentioned in the previous section, the change to the overall algorithm is fairly minimal. This change occurs in the prior function, which now represents a normal distribution (whose parameters were obtained from section 3.3.1). Comparing this method — NS with informed priors — to its uninformed counterpart shows that the estimation is better captured within its distribution, and the ranges are smaller, indicating a more reliable prediction:



Uninformed Prior						Informed Prior					
True Value	Est. Range	Est. Median	Lower	Upper	No. calls	True Value	Est. Range	Est. Median	Lower	Upper	No. calls
0.87	0.78 – 1.00	0.88	0.10	0.12	61,951	0.87	0.81 – 0.97	0.89	0.08	0.08	69,268
0.21	0.12 – 0.32	0.21	0.09	0.11		0.21	0.14 – 0.28	0.21	0.07	0.07	

Figure 3.2: Comparing the posterior distributions generated from informed and uninformed priors

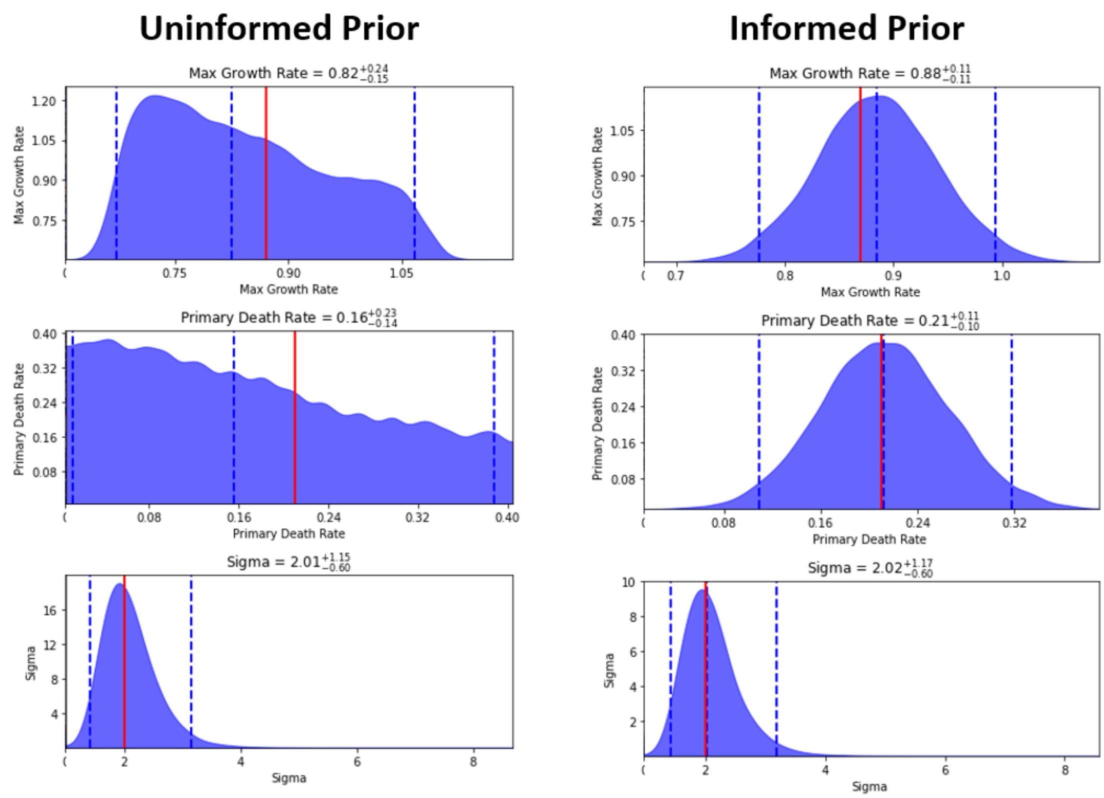
These estimation results were then sampled and used to make predictions on the VCD trajectory. As shown below, the MSE score averaged across all predictions is lower in the new method as compared to the original.



	Area covered by Confidence bands		
Uninformed	909.72	553.89	317.13
Informed	869.31	524.41	300.41

Figure 3.3: Generating VCD predictions, sampled from the parameter estimates

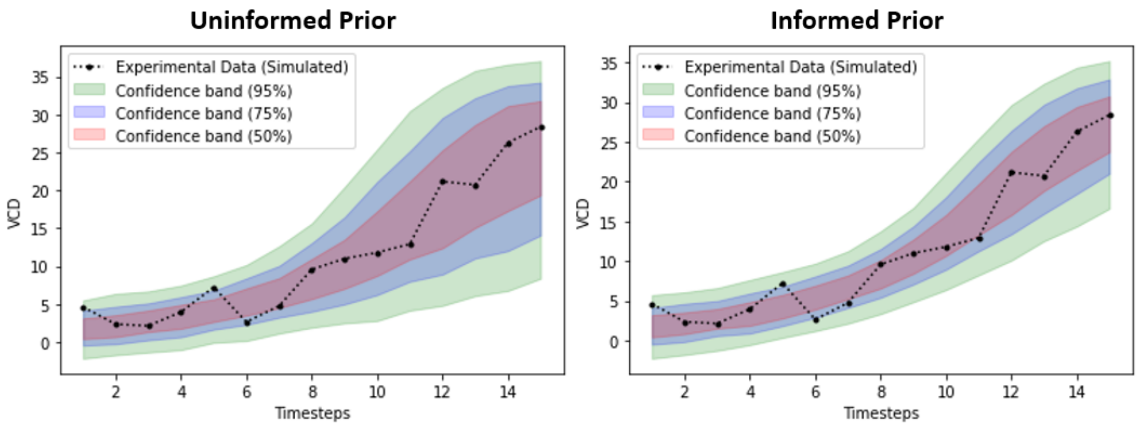
The second scenario is one in which we are trying to estimate the parameters before the run can be completed. In this scenario, the bioreactor has been running for 5 days, as compared to the default 27 days. Comparing the NS results derived from the informed prior method to the original uninformed method, we get the following:



Uninformed Prior						Informed Prior					
True Value	Est. Range	Est. Median	Lower	Upper	No. calls	True Value	Est. Range	Est. Median	Lower	Upper	No. calls
0.87	0.67 – 1.06	**0.82**	0.15	0.24	61,951	0.87	0.77 – 0.99	**0.88**	0.11	0.11	69,268
0.21	0.02 – 0.39	**0.16**	0.14	0.23		0.21	0.11 – 0.32	**0.21**	0.10	0.11	

Figure 3.4: Comparing parameter estimation results for both methods. This scenario has only the first 14 time steps of data available (compared to the conventional ninety-time steps taken during the 27-day period)

The algorithm can estimate the parameters more reliably, as seen by the narrower bounds. Using these estimates to predict trajectories, the informed priors are again able to outperform the default method.



	Area covered by Confidence bands		
Uninformed	244.07	159.07	92.12
Informed	176.62	111.63	65.45

Figure 3.5: Generating VCD predictions, sampled from the parameter estimates (taken from the second scenario)

3.4 The Evidence

The final section of this chapter will be based on the evidence term in Bayesian inference.

$$p(\theta \mid D) = \frac{p(D \mid \theta) \cdot p(\theta)}{p(D)} \tag{3.4.1}$$

In the equation above (Bayes' Theorem), the evidence is the term in the denominator and is defined as the probability of the data being true over the entire range of parameter values for a given model [9]. As mentioned in Chapter 2, for the purposes of parameter estimation, this term is not given much importance. This is because the assumption made is that only one model exists, the one the user currently has. Combined with the fact that this term is hard to calculate means that the evidence is usually given the term "normalizing constant" and neglected. However, this term is also purported to be incredibly useful for other applications, such as model selection [9]. One potential use of model selection in our current framework is in selecting the “right” surrogate model. As this model is in the form of a neural network, the idea is to use this metric to compare between hyperparameter configurations. Determining the model structure that strikes a balance between prediction accuracy and model complexity will help develop surrogate models quicker and with better performance in the future. Looking at similar works, studies that involve the evidence directly are rare to find, especially in the field of process engineering and bioprocesses. There are, however, studies in this field that apply similar model selection techniques. A common metric used in these problems is called the Bayesian Information Criterion (BIC). It works by taking the maximized likelihood value for the parameter and penalizing it by the number of parameters used in the model [47]. However, the BIC is an approximation of the evidence term. It doesn’t factor in the range/shape of the prior space, but just the number of parameters. As mentioned previously, the eventual goal of using the evidence is to select surrogate models for the parameter estimation algorithm. This idea of Hyperparameter learning has been found in other works, but outside the field of bioprocesses and chemical engineering in general. An important example is [48]. In this, a Bayesian-based hyperparameter learning algorithm was developed. The algorithm was run in an online fashion, meaning hyperparameters were adjusted during the model training.

3.4.1 Challenges with calculating the evidence

Another name for the evidence is “Marginalized Likelihood”. This term refers to how the evidence is calculated, through a technique known as marginalization. To illustrate this concept, a derivation for using the evidence in selecting between two models (Model A and B)is given below [9]:

$$\text{posterior ratio} = \frac{p(A \mid D)}{p(B \mid D)} \tag{3.4.2}$$

$$\frac{p(A \mid D)}{p(B \mid D)} = \frac{p(D \mid A)}{p(D \mid B)} \cdot \frac{p(A)}{p(B)} \tag{3.4.3}$$

$$p(D \mid B) = \int p(D, \theta \mid B)\, d\theta \tag{3.4.4}$$

$$p(D \mid B) = \int p(D \mid \theta, B) \cdot p(\theta \mid B)\, d\theta \tag{3.4.5}$$

In 3.4.5, $\int p(D \mid \theta, B)$ refers to the likelihood that the given data matches predictions from model B (essentially the likelihood score). This can be calculated using the marginalization step by integrating over all parameter values. As can be expected, then, a higher number of parameters can very quickly lead to an intractable calculation or one that is too computationally expensive to justify its need. However, some approximations can be made to estimate the evidence more easily. The Nested Sampling algorithm is one such example. This method calculates the evidence first, after which the posterior distribution is obtained. In fact, with every estimation, the evidence score was also provided. Another method is the Laplace approximation, used extensively to select hyperparameters for model selection [48].

3.4.2 Model Comparison — Results

To demonstrate the potential use cases for the evidence, a series of models were built, each describing the same process. The process is a standard continuous stirred-tank reactor (CSTR) undergoing a second-order reaction. The first model had the highest fidelity and was used as the data-generating process (DGP). It uses second-order kinetics and accounts for heat lost to the surroundings. Synthetic measurement noise is also added. The second model was simpler, made using just first-order kinetics. This model also assumed the CSTR to be adiabatic. The third model is considered to be an intermediate model, using first-order kinetics but also accounting for heat loss this time. Process data from the CSTR is provided from the first model. As mentioned above, the NS algorithm automatically provides the evidence score. Therefore, both the second and third models are passed successively through the NS algorithm.

Test Type	ln(Evidence)	MSE Conc 1	MSE Conc 2	MSE Temp
Original Model	+330.570	~0	~0	~0
Simple Model	-210.8395	0.04592	0.000561	10.59867
Simple Model w/ Heat loss	-191.9141	0.03980	0.000622	5.595540

Figure 3.6: Comparing prediction and Evidence score across CSTR models)

The results show that the evidence score (logged) can be a predictor for model performance. The original, high-fidelity model has the highest score, followed by the intermediate model and then the simple one.

3.4.3 Ockham's Razor

Another application of the evidence is Ockham's Razor, the idea that the simplest explanation for something is often the best. The evidence is found to have the ability to select the least complex model that fits the data well. It can penalize a model for having added model complexity over another without the corresponding increase in model performance. To demonstrate this, Model C (the simplest model) is used. A modification was made by adding a "Garbage factor" to it, adding 1 more parameter to this modified version. When passing both the original Model C and its modified counterpart, the evidence score favours the original, simpler model.

Test Type	ln(Evidence)	MSE Conc 1	MSE Conc 2	MSE Temp
Simple Model w/ Heat loss	-191.9141	0.03980	0.000622	5.595540
Simple Model w/ Heat loss + Garbage factor	-198.333	0.03840	0.000671	3.47535

Figure 3.7: Demonstrating Ockham's Razor by comparing the intermediate model with the modified "garbage factor" model)

This leads us to believe that this evidence calculation can choose simpler models over more complex ones if the added complexity doesn't lead to an increase in model performance.

3.4.4 Limitations with the Evidence

Despite the promising results obtained, ultimately it was decided that research into this would be dropped. As mentioned previously, the goal of the evidence for the present work was to select surrogate model configurations for use in the parameter estimation algorithm. However, any hyperparameter algorithm proposed resulted in it being too computationally expensive. This is because of the likelihood computation. In this scenario, the NS algorithm needs to sample different hyperparameter configurations (no. of layers, nodes etc). After a particular configuration is chosen, the Neural network needs to be trained on these parameters. Considering the large number of samples needed to estimate parameters in previous problems, this method is thought to be intractable at the moment.

3.5 Exploring Non-Gaussian Distributions

One of the distinguishing features of MLE methods is their ability to work with a specified noise distribution. This can allow for the estimation of parameters in cases where the underlying measurement noise is not distributed normally. Therefore, in order to demonstrate this feature within the current framework, a simulated test case was created.

A default 3 batch run was passed into the NS algorithm, but this time the synthetic noise generated was sampled from a Skew-Normal (SN) distribution rather than the typical Gaussian distribution. The SN distribution is based on the normal distribution, but has another parameter known as "Skew". This is also known as a shape parameter, as varying this value will affect how skewed the distribution will be. The PDF and derived likelihood function are shown below [49]:

$$\text{PDF} = \frac{2}{\sigma}\phi\left(\frac{x-\mu}{\sigma}\right)\Phi\left(\lambda\frac{x-\mu}{\sigma}\right) \tag{3.5.1}$$

$$l = n\ln\left(\frac{2}{\sigma}\right) - \frac{n}{2}\ln(2\pi) - \frac{1}{2}\sum\left(\frac{x-\mu}{\sigma}\right)^2 + \sum\ln\left(\Phi\left(\lambda\frac{x-\mu}{\sigma}\right)\right) \tag{3.5.2}$$

Where 'ϕ' refers to the PDF of a normal distribution, and 'Φ' refers to the cumulative distribution function (CDF) of the same. For context, the following histogram was generated when 1000 points were sampled from an SN distribution having a skew of 5 and a scale (standard deviation) of 2.

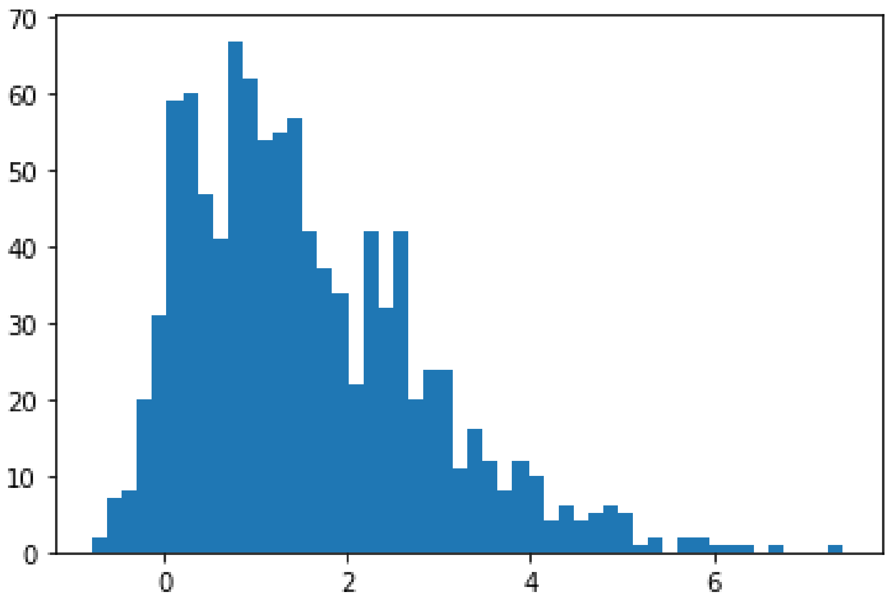
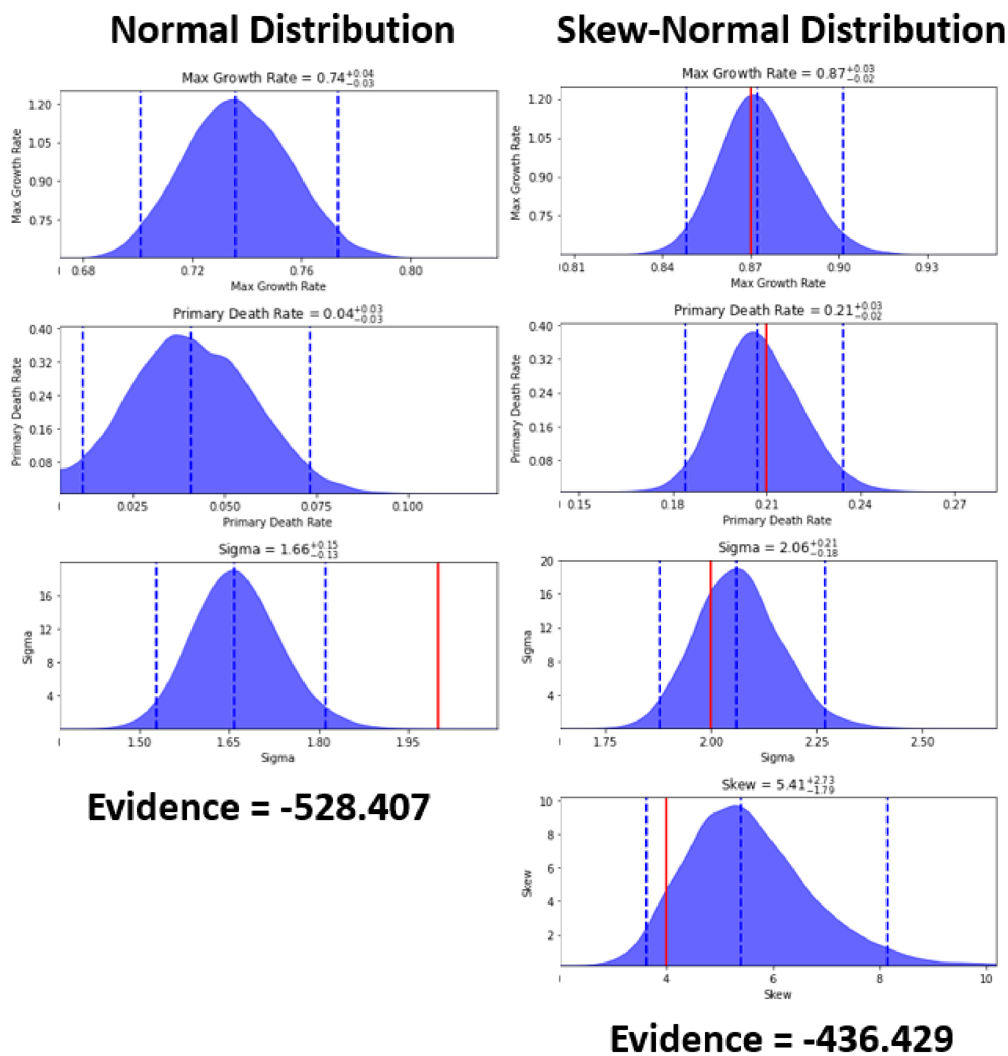


Figure 3.8: 1000 points sampled from an SN distribution with skew = 5 and standard deviation = 2)

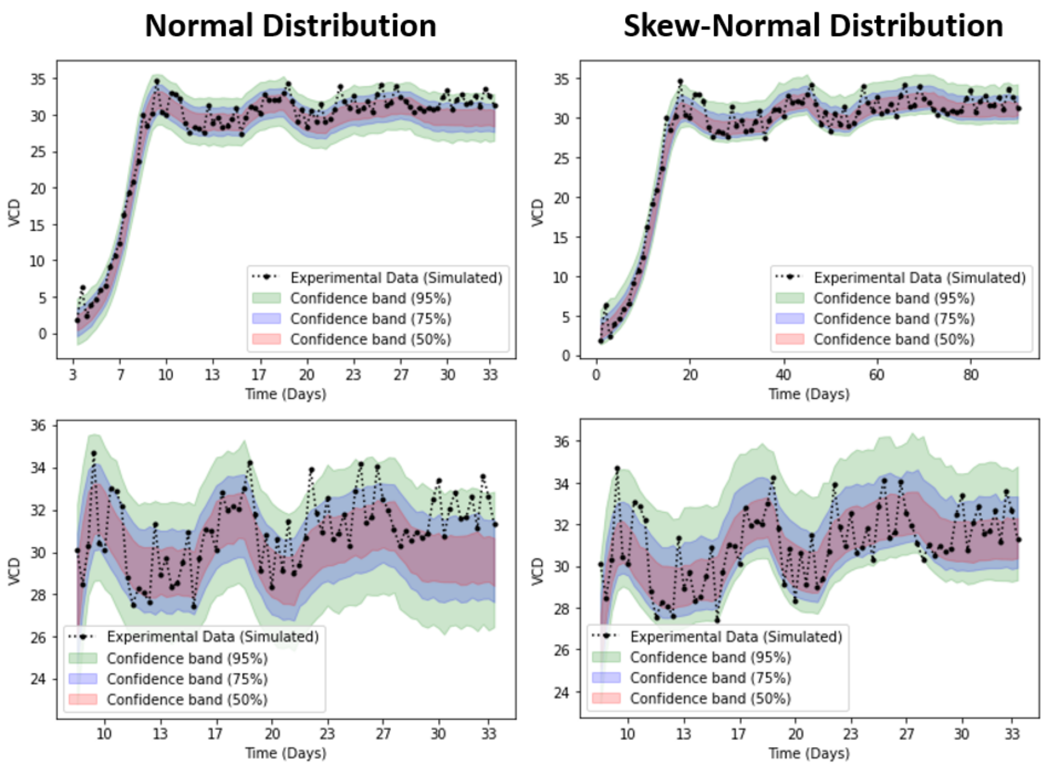
The results from the NS algorithm compare the normal distribution to the SN distribution. The estimates from the normal distribution are not able to capture the true value within its 25th to 75th quantile.



Normal Distribution						Skew-Normal Distribution					
True Value	Est. Range	Est. Median	Lower	Upper	No. calls	True Value	Est. Range	Est. Median	Lower	Upper	No. calls
0.87	0.71 – 0.77	**0.74**	0.03	0.04	64,685	0.87	0.85 – 0.90	**0.87**	0.02	0.03	81,990
0.21	0.01 – 0.07	**0.04**	0.03	0.03		0.21	0.19 – 0.24	**0.21**	0.02	0.03	

Figure 3.9: Comparing parameter estimates using normal and SN distributions)

Using these estimates, a collection of 1000 predictions was sampled and used to construct a confidence band:



	Failure Rate of Confidence Intervals		
Distribution	**CI 95%**	**CI 75%**	**CI 50%**
Normal	5.55	18.88	50
Skew - Normal	2.22	23.33	41.11
	Area covered by Confidence bands		
Normal	598.46	361.27	207.03
Skew-Normal	503.67	309.74	178.99

Figure 3.10: Constructed Confidence bands from normal and SN based parameter estimates

Based on this, the results from the SN distribution had both an overall lower failure rate and a lower area covered by the confidence bands. This indicates that this framework is able to provide better and more reliable results once information about the underlying noise distribution is specified. It is also important to note that the evidence metric reflects directly, by giving the SN distribution a higher value.

Chapter 4

Conclusion

This thesis has explored integrating data-driven techniques with Bayesian inference in order to address practical challenges in parameter estimation problems. The use of surrogate modelling allows for the estimation of parameters from high-fidelity first principles models. The problem of limited data in new processes is addressed through the use of informed priors in the same parameter estimation algorithm. Finally, while an investigation into the evidence term revealed the metric's ability to select the "right" model from a series of alternatives, this method could not be feasibly adapted towards selecting surrogate models without being too computationally expensive. Overall, by demonstrating some of the practical challenges that can be overcome with Bayesian inference, this research has contributed towards the advancement of probabilistic parameter estimation methods in bioprocesses.

4.1 Future Work

Below is a list of future directions this research can take:

4.1.1 Non-Gaussian prior functions

In the work presented in Chapter 3, a normal (Gaussian) distribution was assumed for the distribution obtained using historical data. In the future, it would be more realistic to determine the actual distribution present in the data.

4.1.2 Dealing with "bad" priors

If the prior obtained from historical data has a distribution that is significantly different from the current data, the algorithm has no way of knowing this and will continue to bias the search toward the "wrong" range of parameter values. In the future, it would be an important step to be able to distinguish between a good and a bad prior estimate, perhaps through the use of the evidence score.

4.1.3 Hyperparameter Learning using the Evidence

While the overall consensus in section 3.4.4 was that selecting surrogate models was too computationally expensive, there is still a lot of potential in the evidence term. With the right assumptions made (like the no. of epochs used, or the amount of live points at a particular iteration of the NS algorithm), this method could be used to develop the least complex surrogate models that are still able to predict the data well.

Bibliography

[1] N. Mathias, L. Weir, B. Corbett, and P. Mhaskar, “Leveraging neural networks to estimate parameters with confidence intervals,” *The Canadian Journal of Chemical Engineering*, vol. n/a, no. n/a.

[2] S. Lee, T. F. O’Connor, X. Yang, C. N. Cruz, S. Chatterjee, R. D. Madurawe, C. M. V. Moore, L. X. Yu, and J. Woodcock, “Modernizing pharmaceutical manufacturing: from batch to continuous production,” *Journal of Pharmaceutical Innovation*, vol. 10, pp. 191 – 199, 2015.

[3] S. Sarna, N. Patel, B. Corbett, C. McCready, and P. Mhaskar, “Process-aware data-driven modelling and model predictive control of bioreactor for the production of monoclonal antibodies,” *The Canadian Journal of Chemical Engineering*, vol. 101, no. 5, pp. 2677–2692, 2023.

[4] D. Selişteanu, D. Șendrescu, V. Georgeanu, and M. Roman, “Mammalian cell culture process for monoclonal antibody production: Nonlinear modelling and parameter estimation,” *BioMed Research International*, vol. 2015, p. 1–16, 2015.

[5] C. M. O’Brien, Q. Zhang, P. Daoutidis, and W.-S. Hu, “A hybrid mechanistic-empirical model for in silico mammalian cell bioprocess simulation,” *Metabolic*

Engineering, vol. 66, p. 31–40, July 2021.

[6] D. Şendrescu, S. Tebbani, and D. Selişteanu, *Bioprocesses Parameter Estimation by Heuristic Optimization Techniques*, p. 237–254. Springer International Publishing, 2015.

[7] T. Hernández Rodríguez, C. Posch, J. Schmutzhard, J. Stettner, C. Weihs, R. Pörtner, and B. Frahm, "Predicting industrial-scale cell culture seed trains–a bayesian framework for model fitting and parameter estimation, dealing with uncertainty in measurements and model parameters, applied to a nonlinear kinetic cell culture model, using an mcmc method," *Biotechnology and Bioengineering*, vol. 116, p. 2944–2959, Sept. 2019.

[8] S. S. Jang, H. De la Hoz, A. Ben-zvi, W. C. McCaffrey, and R. B. Gopaluni, "Parameter estimation in models with hidden variables : An application to a biotech process," *The Canadian Journal of Chemical Engineering*, vol. 90, no. 3, pp. 690–702, 2012.

[9] D. S. Sivia, *Data Analysis: A Bayesian Tutorial*. Oxford University PressOxford, June 2006.

[10] *Model Based Parameter Estimation: Theory and Applications*. Springer Berlin Heidelberg, 2013.

[11] S. Pahari, P. Shah, and J. Sang-Il Kwon, "Unveiling latent chemical mechanisms: Hybrid modeling for estimating spatiotemporally varying parameters in moving boundary problems," *Industrial & Engineering Chemistry Research*, vol. 63, no. 3, pp. 1501–1514, 2024.

[12] S. Pahari, P. Shah, and J. S.-I. Kwon, "Achieving robustness in hybrid models: A physics-informed regularization approach for spatiotemporal parameter estimation in pdes," *Chemical Engineering Research and Design*, vol. 204, pp. 292–302, 2024.

[13] S. Jang and R. Gopaluni, "Parameter estimation in nonlinear chemical and biological processes with unmeasured variables from small data sets," *Chemical Engineering Science*, vol. 66, no. 12, pp. 2774–2787, 2011.

[14] S. J. Qin and Y. Liu, "A stable lasso algorithm for inferential sensor structure learning and parameter estimation," *Journal of Process Control*, vol. 107, pp. 70–82, 2021.

[15] J. S.-I. Kwon, M. Nayhouse, G. Orkoulas, D. Ni, and P. D. Christofides, "A method for handling batch-to-batch parametric drift using moving horizon estimation: Application to run-to-run mpc of batch crystallization," *Chemical Engineering Science*, vol. 127, pp. 210–219, 2015.

[16] A. Tran, M. Pont, M. Crose, and P. D. Christofides, "Steam methane reforming furnace temperature balancing using bayesian model identification," in *2018 Annual American Control Conference (ACC)*, pp. 4899–4905, 2018.

[17] J. Yu and S. J. Qin, "Multimode process monitoring with bayesian inference-based finite gaussian mixture models," *AIChE Journal*, vol. 54, no. 7, pp. 1811–1829, 2008.

[18] A. Tulsyan, B. Huang, R. B. Gopaluni, and J. F. Forbes, "On simultaneous on-line state and parameter estimation in non-linear state-space models," *Journal of Process Control*, vol. 23, pp. 516–526, 2013.

[19] R. Chiplunkar and B. Huang, "Modeling and bayesian inference for processes characterized by abrupt variations," *Journal of Process Control*, vol. 127, p. 103001, 2023.

[20] J. Xie, B. Huang, and S. Dubljevic, "Transfer learning for dynamic feature extraction using variational bayesian inference," *IEEE Transactions on Knowledge and Data Engineering*, vol. 34, no. 11, pp. 5524–5535, 2022.

[21] P. Kumari, S. Z. Halim, J. S.-I. Kwon, and N. Quddus, "An integrated risk prediction model for corrosion-induced pipeline incidents using artificial neural network and bayesian analysis," *Process Safety and Environmental Protection*, vol. 167, pp. 34–44, 2022.

[22] M. C. Sadino-Riquelme, J. Rivas, D. Jeison, R. E. Hayes, and A. Donoso-Bravo, "Making sense of parameter estimation and model simulation in bioprocesses," *Biotechnology and Bioengineering*, vol. 117, no. 5, pp. 1357–1366, 2020.

[23] E. J. Koh, E. Amini, G. J. McLachlan, and N. Beaton, "Utilising a deep neural network as a surrogate model to approximate phenomenological models of a comminution circuit for faster simulations," *Minerals engineering*, vol. 170, pp. 107026–, 2021.

[24] A. Shokry Abdelaleem Taha Zied, *A contribution to chemical process operation support: new machine learning and surrogate models based approaches for process*

optimization, supervision and control. PhD thesis, Universitat Politècnica de Catalunya, 2021.

[25] S. Ravindran, P. Singh, S. Nene, V. Rale, N. Mhetras, and A. Vaidya, "Microbioreactors and perfusion bioreactors for microbial and mammalian cell culture," in *Biotechnology and Bioengineering* (E. J. Lopes and L. Q. Zepka, eds.), ch. 7, Rijeka: IntechOpen, 2019.

[26] S. Karra, B. Sager, and M. N. Karim, "Multi-scale modeling of heterogeneities in mammalian cell culture processes," *Industrial & Engineering Chemistry Research*, vol. 49, no. 17, pp. 7990–8006, 2010.

[27] S. R. Eddy, "What is bayesian statistics?," *Nature Biotechnology*, vol. 22, no. 9, p. 111, 2004.

[28] R. van de Schoot, S. Depaoli, R. King, B. Kramer, K. Märtens, M. G. Tadesse, M. Vannucci, A. Gelman, D. Veen, J. Willemsen, and C. Yau, "Bayesian statistics and modelling," *Nature Reviews Methods Primers*, vol. 1, no. 1, 2021.

[29] G. Ashton, N. Bernstein, J. Buchner, X. Chen, G. Csányi, A. Fowlie, F. Feroz, M. Griffiths, W. Handley, M. Habeck, E. Higson, M. Hobson, and A. Lasenby, "Nested sampling for physical scientists," *Nature Reviews Methods Primers*, vol. 2, no. 1, 2022.

[30] K. Hines, "A primer on bayesian inference for biophysical systems," *Biophysical Journal*, vol. 108, no. 9, pp. 2103–2113, 2015.

[31] J. Stone, *Bayes' Rule: A Tutorial Introduction to Bayesian Analysis*. Sebtel Press, 2013.

[32] J. Skilling, "Nested sampling for general Bayesian computation," *Bayesian Analysis*, vol. 1, no. 4, pp. 833 – 859, 2006.

[33] J. S. Speagle, "dynesty: a dynamic nested sampling package for estimating Bayesian posteriors and evidences," *Monthly Notices of the Royal Astronomical Society*, vol. 493, pp. 3132–3158, 02 2020.

[34] E. Higson, W. Handley, M. Hobson, and A. Lasenby, "Dynamic nested sampling: an improved algorithm for parameter estimation and evidence calculation," *Statistics and Computing*, vol. 29, pp. 891–913, dec 2018.

[35] J. Speagle, "A brief introduction to nested sampling," *IPAM's Workshop III: Source inference and parameter estimation in Gravitational Wave Astronomy*, Nov 2021.

[36] G. Cumming and S. Finch, "A primer on the understanding, use, and calculation of confidence intervals that are based on central and noncentral distributions," *Educational and Psychological Measurement*, vol. 61, no. 4, pp. 532–574, 2001.

[37] G. A. Rousselet, C. R. Pernet, and R. R. Wilcox, "The percentile bootstrap: A primer with step-by-step instructions in r," *Advances in Methods and Practices in Psychological Science*, vol. 4, no. 1, p. 2515245920911881, 2021.

[38] Y. Zheng and Z. Wu, "Physics-informed online machine learning and predictive control of nonlinear processes with parameter uncertainty," *Industrial & Engineering Chemistry Research*, vol. 62, no. 6, pp. 2804–2818, 2023.

[39] Z. Wu, H. Durand, and P. D. Christofides, "Safeness index-based economic model

predictive control of stochastic nonlinear systems," *Mathematics*, vol. 6, no. 5, 2018.

[40] A. Mesbah, "Stochastic model predictive control with active uncertainty learning: A survey on dual control," *Annual Reviews in Control*, vol. 45, pp. 107–117, 2018.

[41] M. C. Coleman and D. E. Block, "Bayesian parameter estimation with informative priors for nonlinear systems," *AIChE Journal*, vol. 52, no. 2, pp. 651–667, 2006.

[42] A. Roy, P. Srivastava, and S. Sinha, "Risk and reliability assessment in chemical process industries using bayesian methods," *Reviews in Chemical Engineering*, vol. 30, no. 5, pp. 479–499, 2014.

[43] L. Luo, Y. Yao, and F. Gao, "Bayesian improved model migration methodology for fast process modeling by incorporating prior information," *Chemical Engineering Science*, vol. 134, pp. 23–35, 2015.

[44] T. Hernández Rodríguez, C. Posch, J. Schmutzhard, J. Stettner, C. Weihs, R. Pörtner, and B. Frahm, "Predicting industrial-scale cell culture seed trains–a bayesian framework for model fitting and parameter estimation, dealing with uncertainty in measurements and model parameters, applied to a nonlinear kinetic cell culture model, using an mcmc method," *Biotechnology and Bioengineering*, vol. 116, no. 11, pp. 2944–2959, 2019.

[45] L. A. Gibson and K. B. McAuley, "Bayesian parameter estimation using truncated normal distributions as priors for parameters in fundamental models of

chemical processes," *The Canadian Journal of Chemical Engineering*, vol. n/a, no. n/a.

[46] S. Sarna, N. Patel, B. Corbett, C. McCready, and P. Mhaskar, "Determining appropriate input excitation for model identification of a continuous bio-process," *Digital Chemical Engineering*, vol. 6, p. 100071, 2023.

[47] H. Wang, C. Kontoravdi, and E. A. del Rio Chanona, "A hybrid modelling framework for dynamic modelling of bioprocesses," in *33rd European Symposium on Computer Aided Process Engineering* (A. C. Kokossis, M. C. Georgiadis, and E. Pistikopoulos, eds.), vol. 52 of *Computer Aided Chemical Engineering*, pp. 469–474, Elsevier, 2023.

[48] A. Immer, M. Bauer, V. Fortuin, G. Rätsch, and M. E. Khan, "Scalable marginal likelihood estimation for model selection in deep learning," 2021.

[49] A. Yalçınkaya, B. Şenoğlu, and U. Yolcu, "Maximum likelihood estimation for the parameters of skew normal distribution using genetic algorithm," *Swarm and Evolutionary Computation*, vol. 38, pp. 127–138, 2018.