NOISE AWARE BAYESIAN PARAMETER ESTIMATION IN BIOPROCESSES

NOISE AWARE BAYESIAN PARAMETER ESTIMATION IN BIOPROCESSES: USING NEURAL NETWORK SURROGATE MODELS WITH NON-UNIFORM DATA SAMPLING

By LAUREN WEIR, B.ENG.BIOSCI

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:	Noise Aware Bayesian Parameter Estimation in Bioprocesses: Using Neural Network Surrogate Models With Non-Uniform Data Sampling
AUTHOR:	Lauren Weir B.Eng.Biosci, McMaster University, Hamilton, Canada
SUPERVISOR:	Dr. Prashant Mhaskar
NUMBER OF PAGES:	xii, 43

Lay Abstract

Bioprocesses require models that can be developed quickly for rapid production of desired pharmaceuticals. Parameter estimation is necessary for these models, especially first principles models. Generating parameter estimates with confidence intervals is important for model based control. Challenges with parameter estimation that must be addressed are the presence of non-uniform sampling and measurement noise in experimental data. This thesis demonstrates a method of parameter estimation that generates parameter estimates with credible intervals by incorporating measurement noise in experimental data, while also employing a dynamic neural network surrogate model that can process non-uniformly sampled data. The proposed technique implements Bayesian inference using nested sampling and was tested against both simulated and real experimental fed-batch data.

Abstract

This thesis demonstrates a parameter estimation technique for bioprocesses that utilizes measurement noise in experimental data to determine credible intervals on parameter estimates, with this information of potential use in prediction, robust control, and optimization. To determine these estimates, the work implements Bayesian inference using nested sampling, presenting an approach to develop neural network (NN) based surrogate models. To address challenges associated with non-uniform sampling of experimental measurements, an NN structure is proposed. The resultant surrogate model is utilized within a Nested Sampling Algorithm that samples possible parameter values from the parameter space and uses the NN to calculate model output for use in the likelihood function based on the joint probability distribution of the noise of output variables. This method is illustrated against simulated data, then with experimental data from a Sartorius fed-batch bioprocess. Results demonstrate the feasibility of the proposed technique to enable rapid parameter estimation for bioprocesses.

Acknowledgements

I would like to first thank my supervisor, Dr. Prashant Mhaskar. I am so grateful for the opportunity to have been a part of your research group. Your continued guidance and patience was an asset to me in completing my degree, and the weekly meetings always made me more confident in the work I was doing. To Dr. Brandon Corbett, thank you for seeing my potential and being a wonderful mentor to me, ever since our industry capstone collaboration. I'm grateful for my experience at Sartorius and looking forward to the next stage of my career. To my colleague Nigel, it was great collaborating with you on our projects, and thank you for your help with the conference presentations. Thank you to Dr. Jake Nease, it's been a pleasure being a TA for your course these past couple of years, and I wish you all the best in your new position.

To the Mhaskar research group and the friends I've made in the Chemical Engineering Department, thank you for the support and making my years here memorable. Thank you to McMaster University for being my home away from home for 7 years, and being a welcoming community for both my undergrad and grad degrees. Thank you to Sartorius and the MACC for this collaboration and providing me the opportunity to learn more about industry work and share my own work with others.

Finally, I would like to thank my parents, for their continued support and always

believing in me, and reminding me that I'll always have a place to come home to. To my extended family, aunts, uncles, cousins, friends, and my grandmothers, thank you for always cheering me on, no matter how far we are from each other. And to my younger sister, thanks for always encouraging me to speak my mind and stand up for myself. May we have many more years navigating life together.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation

The mathematical understanding of bioprocesses remains a topic of interest in the biopharmaceutical industry. As global sales in the industry increase, the demand for monoclonal antibodies is dominant [11], driving the need for rapid and optimal production for these particular processes. The underlying complexity of bioprocesses as well as high experimental costs encourages the use of models that can be developed quickly and reliably [21], to in turn be used for process design, operation and control. Several approaches have been used to model bioprocesses. First principles models are often used to describe many processes, however there is limited development in these these models for the complex biological systems in cell cultures, where the mechanisms are not yet well understood [11][21]. Employing the use of data-driven modelling presents its own challenges, mainly the requirement of sufficient experimental data, which is often limited due to production costs and resource availability [19][2], and the heterogeneity from the resulting data, consisting of online and offline measurements

and varying sampling times [11]. To improve model accuracy, hybrid models have been introduced by combining first principles with experimental data [11][31].

When building these models, especially first principles models, estimating model parameters is required. Several studies have demonstrated different techniques to estimate parameters for bioprocesses. In one result [22], genetic algorithm was applied to estimate parameters for a plant cell culture system using experimental data. A study of a Chinese Hamster Ovary (CHO) cell bioprocess determined maximum growth rate and maximum death rate experimentally with differential analysis methods, yield ratios with linear regression, and minor parameters with differential evolution, applying the values to a first principles model based on Monod kinetics [5]. Particle swarm optimization was used in another study to estimate parameters for experimental data of a similar bioprocess described by a set of differential equations, solving the issue of ODE stiffness [28]. Another result [26] proposed starting with a simplified model with a small subset of parameters. These parameters would be estimated by solving an optimization problem, and then substituted into a more complex model, gradually building up towards the original model, and this was tested on simulated data. In addition, a fairly recent contribution applied nonlinear regression to a bioreactor kinetic model and generated confidence intervals for the parameters using triplicates of experimental data [25]. Finally, a study compares techniques for dynamic parameter estimation, showing that moving horizon estimation outperforms weighted least squares estimation for experimental data [10].

While many recent contributions used a deterministic approach for parameter estimation, a few studies have taken a probabilistic approach, which is meaningful in

that it allows the use of prior information and stochastic exploration of the parameter space [37]. Bayesian inference has been utilized for parameter estimation for a kinetic cell culture model based on Monod kinetics [9]. The model consisted of six algebraic equations and seven differential equations that were numerically solved in MATLAB. The Markov chain Monte Carlo (MCMC) method was used to compute the posterior parameter distributions. Bayesian updating is then performed by using the obtained posterior distributions as the new prior distributions and using the MCMC method again with new experimental data. The resulting prediction bands are plotted against the state variables from test data, observing how many points fall within the bands . Metropolis-Hastings MCMC algorithm has also been deployed for parameter estimation using the genetic regulatory network and JAK–STAT signal transduction pathway as biological process examples, showcasing its effectiveness on small datasets [13]. Another study applied the MCMC method for parameter estimation with a kinetic model with six differential equations, using the resultant proposed parameters to predict the variables of interest at subsequent sampling times [39].

In addition, Bayesian techniques have been used in a multitude of process control applications. One result used Bayesian model identification to determine a model for the outer tube wall temperature of a steam methane reforming furnace [34]. Kinetic Monte Carlo simulation model of a batch crystallization process has been developed and utilized in a model predictive controller [18]. To address issues in multimode processing monitoring, finite Gaussian mixture model for data clustering has been utlized with Bayesian inference to calculate the posterior probabilities of samples for the Gaussian components [40]. Another contribution remarks on the use

of Bayesian Lasso when developing a more stable lasso algorithm to apply to an inferential model of a chemical plant [24]. The Bayesian approach has also been used in an algorithm for on-line parameter estimation as well as state estimation with a sequential-importance-resampling filter and introducing kernel smoothing [35]. Another contribution [17]paired an artificial neural network and Bayesian analysis to predict the risk of pipeline corrosion. The neural network received significant attributes of the incident as input and produced the cause and resulting consequences as output, while Bayesian analysis, when provided the cumulative time to an incident used the MCMC algorithm to determined the probability of the incident. Variational Bayesian inference has been utilized to estimate the parameters of a latent variable model that captures abrupt and regular variations in datasets [3]. Another application of variational Bayesian inference has been to estimate parameters of dynamic models involved in a transfer slow feature analysis for the purpose of soft sensor modelling [38].

Methods used for Bayesian inference such as MCMC [39] and nested sampling [30] require calculations from the model using the proposed parameters, and this can increase computational costs if the model is complex. Neural networks are a potential solution to this issue, as feedforward networks are considered "universal approximators" [12] and have been combined with first principles models, reducing computation time [23].

Surrogate models using neural networks have been developed from simulations such as a fluidized bed reactor [4] and units in a biodiesel production plant [7]. However, most of these results discuss preventing overfitting due to a sparse dataset [16] or simulating a sufficient amount of scenarios being computationally expensive [14].

There has not been much discussion on the topic of using neural networks for surrogate modelling where data can be generated as needed to avoid overfitting altogether, while also accurately replicating model predictions, and thus enabling use within parameter estimation algorithms. Another key tool missing from existing literature is the ability to build dynamic NN models for instances such as bioreactor data with measurements where the measurement frequency is non-uniform, thus different from variable to variable, and also potentially different from one sampling instance to another. Some relevant approaches include designing a convolutional neural networks to concatenate variables along the time interval for crowd flow predictions [41] and interpolate between non-uniform input samples in signal processing [29]. When processing biomedical data for body sensors, the non-uniform sampling schedule was addressed by including the time delay between coordinates as input to the neural network [20]. A neural network structure for dynamic modeling of process data with non-uniform sampling remains unavailable.

Motivated by the above, this thesis proposes a method of parameter estimation using the nested sampling algorithm for Bayesian inference, with a neural network as a surrogate model. The Neural Network is designed in a way that enable training and prediction of variable at non-uniform measurement frequency. The method is applied to a model for a CHO cell culture in a fed batch bioreactor using real experimental data to estimate the maximum growth rate and primary death rate parameters in a first principles model.

1.2 Outline of the Thesis

Chapter 2 outlines the background of the bioprocess as well as the concepts and theories of Bayesian inference. Chapter 3 describes the development and application of the method against real experimental data. Results are described in Chapter 4 and conclusions are presented in Chapter 5.

Chapter 2

Preliminaries

The bioprocess under consideration and the nature of the data collected is described, along with the hybrid state-space model (HSSM), for which the parameters need to be estimated. The concepts of Bayesian Inference are discussed and the Nested Sampling Algorithm is introduced.

2.1 Fed-batch Bioreactor process

The experimental data of 12 batches for a design of experiments from the Sartorius AMBR 250 ® bioreactor system was provided by Sartorius for this work. Each batch was run in a 250 mL vessel (see Figure 2.1 below for a schematic) for approximately 12 days in fed-batch mode, with periodic bolus additions to maintain adequate glucose levels for cell growth. 8 of these batches experienced a change in setpoint temperature and/or pH implemented on the 7th day of the experiment, which was around the beginning of the stationary phase of the cell culture. The remaining 4 batches served as control. The objective is to estimate parameters for a first principles model

developed to describe this process.

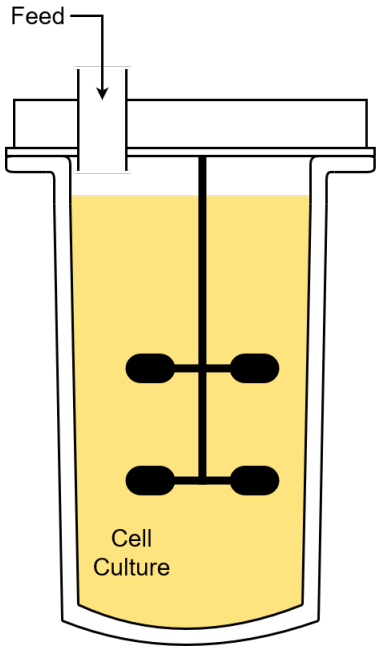

Figure 2.1: Diagram of 250 mL vessel from the Sartorius AMBR 250 ® bioreactor system.

2.2 Experimental Data Description

The recorded dataset for each batch contained process, feed, and metabolite data, which was measured with varying sampling times either by the AMBR system itself, or by the Nova Biomedical BioProfile® FLEX2. For the metabolite data (eg. glucose, viable cell density), the age of each batch was recorded in days, with 35 samples per batch. The process data (eg. temperature, pH) was recorded at the same time as the metabolite data. The time intervals between each sample are non-uniform. The feed data (eg. glucose additions) for each batch was also recorded in days, however there were 53 samples per batch. While this data was recorded at the same time as

the rest of the set, the 18 extra points accounted for the time points when feed was added. Data used for this work was selected based on its relevance to the growth of the cell culture. The sequence of variables included the time stamp, temperature, pH, glucose bolus additions, glucose concentration, VCD, dead cell density, lysed cell density and biomaterial.

2.3 Hybrid State Space Model

The hybrid state-space model (HSSM) provided by Sartorius is an example of a first principles based model, containing 10 ODEs and 4 parameters (provided by the user) describing cell metabolism.

Three of these equations are shown below as examples [27]:

$$\frac{dX_v}{dt} = (\mu_{max} - u_d)\, X_v \tag{2.3.1}$$

$$\frac{dX_d}{dt} = u_d X_v - k_l X_d \tag{2.3.2}$$

$$\frac{dX_l}{dt} = k_l X_d \tag{2.3.3}$$

where X_v, X_d, and X_l represent the viable cell density, dead cell density, and lysed cell density, respectively. μ_{max} is the maximum growth rate, u_d is the primary death rate, and k_l is the lysing rate. The HSSM includes balances on the metabolites, with appropriately defined functions capturing the interdependence of variables.

The model is coded in Python, and works by initializing a reactor object and parameters such as maximum growth rate, primary death rate, lysing rate, and toxicity rate of the cell culture are provided by the user. Optional conditions such as the reactor temperature and pH can also be provided to the model, as well as glucose feed bolus additions. The reactor's initial state is set up using this information, and can be run for a specified time by solving the first principles equations that describe cellular metabolism.

The main objective for this work is to estimate the maximum growth rate and primary death rate of the cell culture using the proposed method and analyze the output of the glucose concentration and the viable cell density (VCD) after using the estimates in the HSSM, as well as to illustrate the ability to account for the measurement noise distribution in the parameter estimation.

2.4 Bayesian Inference

Bayesian statistics are based on the concept of Bayes Theorem, as shown below [15, 36]:

$$P(\theta \mid D) = \frac{P(D \mid \theta) \cdot P(\theta)}{P(D)} \tag{2.4.1}$$

where the prior ($P(\theta)$) is the probability distribution of the parameters θ independent of observed data D (ie. prior knowledge of the parameter values before data is measured). Likelihood ($P(D \mid \theta)$) is the probability distribution of the data D given the parameters θ [36]. In other words, it determines the likelihood of the parameter values given the data. Posterior ($P(\theta \mid D)$) is the probability distribution of the

parameters θ given the data D. Evidence ($P(D)$) is the probability distribution of the observed data D, and is used as a normalization constant.

In general, the prior refers to what is known about the parameters before measuring the data, and the likelihood function refers to obtained knowledge of these parameters given the observed data. The product of the likelihood and the prior is proportional to the posterior, which refers to the updated knowledge about the parameters once the data is provided [36].

The theorem can be further simplified as follows [1]:

$$P(\theta) = \frac{\mathcal{L}(\theta) \cdot \pi(\theta)}{\mathcal{Z}} \tag{2.4.2}$$

$$\mathcal{Z} = \int \mathcal{L}(\theta)\pi(\theta)\, d\theta \tag{2.4.3}$$

The evidence integral is difficult to estimate since θ could potentially have several dimensions depending on the number of parameters [30][1]. The nested sampling algorithm addresses this issue in the next subsection.

2.5 Nested Sampling

In nested sampling, the integral is converted from the multidimensional parameter space to a one dimensional likelihood space [30][1][32]:

$$dX = \pi(\theta)d\theta \tag{2.5.1}$$

$$X(\lambda) = \int_{\mathcal{L}(\theta)>\lambda} \pi(\theta)\, d\theta \tag{2.5.2}$$

where X is the prior mass over the parameter space where the likelihood is equal, and X(λ) is the cumulative prior mass where the likelihood function is greater than some value λ. Taking the inverse [30][1]:

$$\mathcal{Z} = \int_0^1 \mathcal{L}(X)\, dX \tag{2.5.3}$$

This equation depicts the likelihood function as a positive and decreasing integrand with an area of Z, which is estimated as a weighted sum as follows [30]:

$$\mathcal{Z} = \sum_{i=1}^{m} w_i \mathcal{L}_i \tag{2.5.4}$$

where w_i is the Trapezoidal rule applied to the integral over $\mathcal{L}(X)$. The likelihood function can be approximated as the joint probability of each data point given some parameter value, as described below [36][8]:

$$\mathcal{L}(\theta) \approx \prod_{i=1}^{n} p(y_i \mid \theta) \tag{2.5.5}$$

where $p(y_i \mid \theta)$ refers to the probability of data point y_i given parameter(s) θ. This allows for measurement noise distributions present in data to be accounted for when the algorithm is applied.

In the nested sampling algorithm, some predefined number (N) of live points are sampled from the prior. For each iteration, the likelihood of each point is calculated

and the point with the lowest likelihood is replaced with one of higher likelihood until convergence. This is described in detail as follows [30]:

1. Sample the prior N times to get points $(p_1, ..., p_N)$ and calculate their likelihood (Initially: $Z = 0, X_0 = 1$)

2. For $i = 1, 2, 3, ...j$

   $L_i := \min(\text{current likelihood values})$

   $X_i := e^{\frac{-i}{N}}$

   $w_i := X_{i-1} - X_i$ or $\frac{1}{2}(X_{i-1} - X_{i+1})$

   $Z := Z + w_i L_i$

   Replace the point with the lowest likelihood with a new point with a likelihood greater than L_i in proportion with prior $\pi(p)$

3. Return Z

The algorithm involves several calls to the likelihood function, which in turn involves computing the output, this may become computationally expensive if one were to use a complex first principles model in this computation. The intent of using a surrogate model is to reduce the computation time of the overall algorithm. Note that the intent of the surrogate model is not to model the process directly, but simply to replicate the first principles model, and more importantly, capture the effect of the parameters on the predictions generated by the first principles model.

Chapter 3

Proposed NN-surrogate Model

This chapter describes the proposed method. The design of the neural network surrogate model is outlined, presenting an algorithm tailored for surrogate dynamic models, allowing for non-uniform sampling. The internal structure of the neural network is explained, as well as the process of training and testing. Finally, the setup of the nested sampling algorithm is discussed.

3.1 Neural Network Design

In this thesis, a neural network is used as a surrogate model in place of the HSSM to address the computational challenges associated with use of first principles models in Dynamic Nested Sampling. The model is meant to be used in place of the HSSM to reduce computational load when implementing nested sampling, therefore it was trained and tested with simulated data generated from the HSSM. The simulated data was generated to be similar to the experimental data, with the difference being that the lysed cell density and biomaterial content are calculated outputs in the first

principles model, and were therefore used in the surrogate model. Note that these are not part of the measured variables in the experimental data set- thus, when eventually being used as part of the nested sampling algorithm, these variables do not constitute a part of the likelihood function. The other key distinction is that the surrogate model is being designed so that it is able to predict the variables predicted by the HSSM for different values of the the growth rate and the death rate. Therefore, when generating training data for the surrogate modeling, the data sets included variations in the growth rate and death rate, and included these as input variables to the model.

The neural network structure was designed so the input data contained the sampling time interval from time t_0 to time t_1, the temperature at t_0, the change in temperature during the interval $(t_1 - t_0)$, the pH at t_0, the change in pH during the interval $(t_1 - t_0)$, the glucose feed addition, the glucose level in the reactor, the viable cell density (VCD), the dead cell density, the lysed cell density, the biomaterial content, the maximum growth rate and the primary death rate all at t_0. The output data contained the glucose level in the reactor, the VCD, the dead cell density, the lysed cell density, and the biomaterial all at t_1. The main variables of interest were the VCD and the glucose levels, which are affected by the growth rate and death rate. The data was arranged such that the input data went from t_0 to the second last time point t_{f-1} and the output data went from t_1 to t_f. This is shown in Figure 3.1 below.

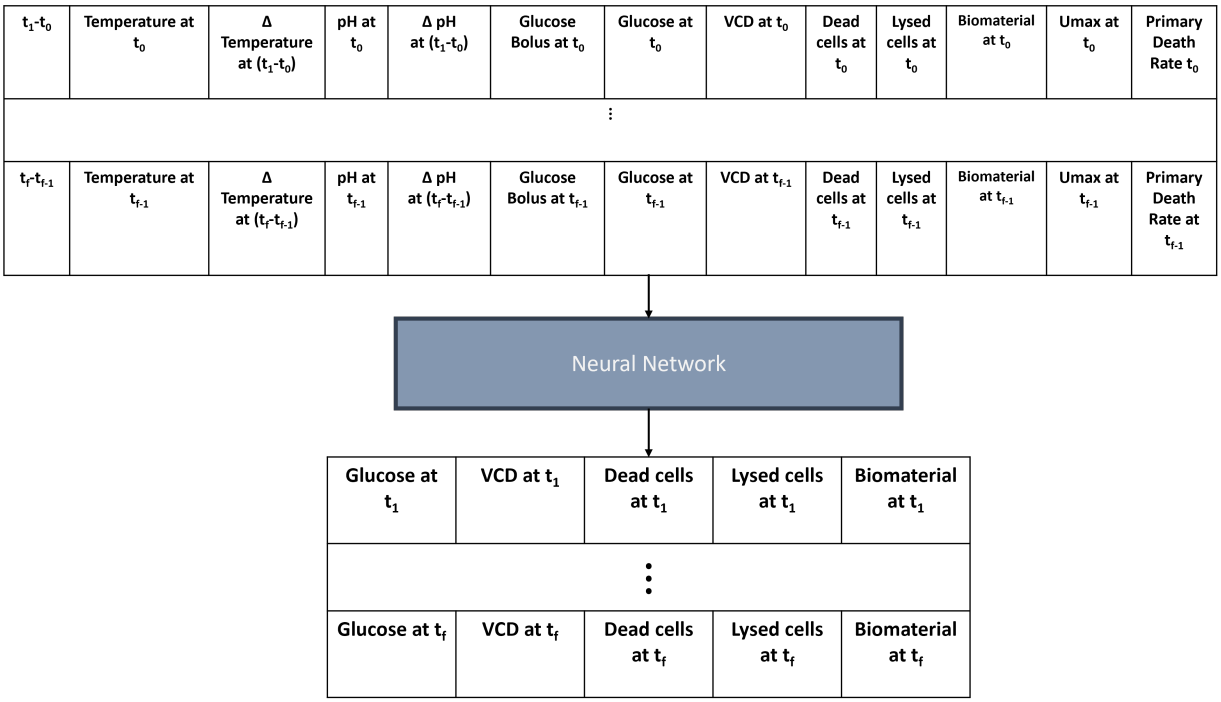

Figure 3.1: Surrogate model structure.

Remark 1 *One of the issues that has not been explicitly addressed in the use of NN for dynamic modeling is to allow for non-uniform sampling intervals. When the sampling intervals are uniform, and consistent across all data, the resultant NN dynamic model has the sampling time/discretization time step in-built. In the present case (as is the case with several processes), the variables are not measured at uniform intervals. One could of course assume the uniformity of the time interval, and that would introduce a further mismatch between the model and the data. In contrast, the present manuscript demonstrates how to enable the NN model to be built to predict the outputs for non-uniform sampling by including the sampling interval as part of the training block. In essence, the resulting NN model 'surrogates' the first principles model ODEs for various discretization steps, eventually accomodating for data that is collected at non-uniform sampling intervals.*

Remark 2 *Several other options were considered before utilizing the time interval, and are discussed next to better illustrate the reasoning behind the eventual data arrangement. One was to use the time stamps them selves- with the current time step as part of the X block and the next time step as part of the Y block. Such a data arrangement incorrectly suggests time interval being accounted for in the training. As a matter of fact, such a choice would be completely incorrect. In essence, it would require the NN model to be trained to 'predict' the sampling interval, which is simply unnecessary, and would degrade the NN performance. On the other hand, if the current time stamp was used as part of the X block, but not used as the Y block, it would essentially 'ask' the NN to model a time varying nonlinear system, but with a uniform sampling interval, which is not the intent either.*

Remark 3 *Another key choice that was made in the design of the NN was to include not just the current values of the pH and temperature, but also any step changes in temperature and pH being implemented. This is important as, in the absence of this information, the NN model would not 'know' that a pH change or temperature changes has happened, and to attribute the change in other variables to changes in pH and temperature. Removing this information essentially prevents the design of an appropriate surrogate model. It is also important to recognize that the use of these variables as part of the input is consistent with the notion that the model is not meant to predict the pH and temperature evolution, but instead, given a pH and temperature profile, predict the evolution of the other variables of interest.*

3.2 Neural Network Training and Testing

The other key contribution of the present manuscript is the recognition that a surrogate model is being developed, and hence the training approach should be different from what is traditionally used when training an NN against experimental data. In order to develop an optimal neural network surrogate model, the following algorithm is proposed and implemented:

1. Identify relevant input/output variables
2. Generate sufficient dataset for training and testing the network
3. Scale the data and add input perturbations if possible
4. Perform hyperparameter search (Use results as a baseline, more layers and nodes can be added if needed)
5. Implement the techniques below when training to improve the model's performance:
   (a) Learning rate decay
      i. Apply a learning rate scheduler and implement a small learning rate decay
      ii. Schedule the decay for when the training loss does not improve after a certain number of epochs
      iii. Assess model predictive performance and adjust the decay and/or the epoch threshold
   (b) Validation set

i. Create a validation set by taking data from the training set

ii. For each epoch, calculate the validation loss and use it to determine whether the model should be saved

iii. Assess model predictive performance and adjust the amount of provided validation data

(c) Early stopping

i. Terminate training if the validation loss does not improve after a certain number of epochs

ii. Assess model prediction performance and adjust the epoch threshold

6. Assess the overall predictive performance

The algorithm allows for the development of a neural network that encourages increasing the number of nodes and layers while also employing techniques that improve the network's predictive ability to replicate the first principles model. The key is to use the first principles model to generate as much data as necessary to avoid overfitting. In some cases, models have a large computational cost[14], but the time requirement can be ameliorated by using tools like parallel processing. One of the key contributions of the algorithm is not so much as what to do, but more of what not to do, i.e., when building surrogate models to not use dropout [14] and/or weight decay [33].

Remark 4 *The provided algorithm is meant to develop a surrogate model that replicates the first principles model as closely as possible. While strategies such as dropout and weight are effective when avoiding overfitting, the main objective of this work's surrogate model is to be an 'exact' replicate of the first principles model. Dropout*

and weight decay are used to generalize the neural network for better prediction of unseen, noisy data. However, the approach here is to provide the neural network with a sufficient amount of data from simulation such that it covers the intended range of the first principles model.

For training and testing the surrogate model, the HSSM was used in Python to generate 5000 batches of simulated data, in which each batch simulates a fed batch scenario run for 12 days as presented in the actual experimental data. Each batch used a set of sampling times and feed bolus addition schedule from a randomly selected batch from the experimental data. A randomized combination of the two parameters of interest (ie. maximum growth rate and primary death rate), temperature, and pH were also provided and scheduled to change at 20 random time points throughout the run to introduce input perturbations. Both the cell lysing rate and the toxicity rate were kept constant.

The neural network was developed in the *Pytorch* package with its structure and hyperparameters determined following the proposed algorithm. The network had 10 hidden layers and 500 nodes, using the ReLU function for the hidden layers and the sigmoid function for the final output layer. The 5000 batches were divided into 4497 training, 500 validation, and 3 testing sets, then scaled such that inputs were between -1 and 1, and outputs were between 0.2 and 0.8. During training, the input and output sets were provided in their entirety. Training was performed via the back propagation algorithm using the Adam optimizer with a mini-batch size of 52 and an initial learning rate of 1e-4 with exponential learning rate decay of 0.99. The calculated validation loss was also taken into account during training.

During testing, the first row of input was provided to the neural network with

the predicted output being provided to the next row of input to reflect a scenario in which only the first data point is known. The results of the recursive prediction are plotted against the true values in Chapter 4 to determine the predictive ability of the neural network.

Remark 5 *The neural network must use previous outputs as subsequent inputs during testing. This way, only the initial conditions of the process need to be known. For example, if the data at time t_0 is known, the neural network will use it to predict the output data at time t_1. Subsequently, this data at t_1 is used to predict the output data at time t_2. If instead the neural network was provided previous output from the HSSM to test for only one time step ahead prediction, this surrogate model would not be useful as part of the nested sampling algorithm because not only does the surrogate model become dependent on the HSSM, the HSSM would also have function calls from the method to provide the neural network with data, negating the whole purpose of using the NN surrogate model.*

3.3 Setting up Nested Sampling

To preprocess the data for use in this work, the relevant metabolite, process, and feed data were lined up such that the sample times matched, and interpolated at sample times where feed was added. The lysed cell density and biomaterial were not recorded, thus these variables were not used as part of the likelihood function. The objective of the nested sampling was to determine the maximum growth rate and primary death rate.

The input data to the nested sampling had 13 columns of variables containing

the time intervals various variables measured before the interval, while the output data had 5 columns of variables containing the measured variables after the interval. The sampling intervals are obtained from the experimental data by subtracting the recorded age at the time of one sample from the recorded age of the next sample, creating 52 data points for one batch of data.

The nested sampling algorithm for the use of the proposed method was developed in Python using the *dynesty* package [32]. The neural network surrogate model was incorporated in the algorithm at the likelihood calculation step. As explained previously, the likelihood can be expressed as the joint probability of the observed data.

A simple and commonly utilized and often practical situation is where the measurement noise can be assumed to have a normal distribution, which can be expressed in the likelihood function as follows:

$$\mathcal{L}(\theta) = \prod_{i=1}^{N} \frac{1}{\sigma\sqrt{2\pi}} e^{\frac{-1}{2}(\frac{y_i - \hat{y}_i}{\sigma})^2} \tag{3.3.1}$$

Taking the logarithm to make results easier to work with:

$$\mathcal{L}(\theta) = \frac{-n}{2}\ln(2\pi) + \frac{-n}{2}\ln(\sigma^2) - \frac{1}{2\sigma^2}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 \tag{3.3.2}$$

where y_i is the observed data and $\hat{y}_i$ is the model calculated data for a particular θ.

Considering the complexities of measuring data in some instances, it may be that the measurement noise is not normally distributed. This requires a function that can accommodate various noise distributions so the likelihood calculations of parameter estimates for real data are as accurate as possible. For example in this work, VCD

is measured via the Trypan Blue/Digital Imaging method, where the error is larger for larger measurements. This behavior can be handled by adjusting the likelihood function such that the standard deviation changes for each output point of the VCD.

This change was implemented as shown below:

$$\sigma_{VCD} = \sigma \frac{y_{i,VCD}}{MAX(y_{i,VCD})} \tag{3.3.3}$$

where σ is a predefined standard deviation, $y_{i,VCD}$ is the current VCD at time interval i, and $MAX(y_{i,VCD})$ is the maximum VCD (0.8 due to scaling).

Remark 6 *It could be argued that ordinary least squares could be used in place of a likelihood function. However, maximum likelihood estimation is more general and can be used for distributions such as normal, binomial, and Poisson for statistical models. This allows for more complex distributions to be utilized. Ordinary least squares is a special case of maximum likelihood; if the log-likelihood function is taken of a linear regression model with errors normally distributed and a constant variance σ^2, this results in an ordinary least squares estimate [6].*

The nested sampling algorithm makes several calls to the likelihood function depending on the size of the problem, therefore there are several calls to the model generating $\hat{y}_i$ for different guesses of θ. Note that this is the reason that replacing the HSSM with the neural network allows better handling of the computational load.

Remark 7 *The processing time of the nested sampling algorithm was compared when it was paired with the neural network surrogate model to when it was paired with the HSSM. When paired with the surrogate model, processing time was approximately 13 minutes. When paired with the HSSM, the code ran for approximately 13 minutes then*

crashed. The nested sampling algorithm provides the model with the candidate growth rate and death rate parameters, however since the HSSM contains complex ODEs, poor parameter values can cause numerical solvers to take infinitesimally small step sizes, resulting in error. The surrogate model does not contain ODEs, and thus processes the data points faster than the HSSM, had the HSSM been able to continue.

The algorithm runs as described in Chapter 2 and calculates the likelihood of each parameter prediction using the neural network results. The *dynesty* package can then produce a visual representation of the resultant marginalized posterior probabilities of the parameters.

Chapter 4

Surrogate Modeling and Parameter Estimation Results

This chapter first illustrates the surrogate modeling results, i.e., the ability of the NN to replicate the HSSM. The parameter estimation results are next demonstrated using simulated data first, and finally using data from experiments.

4.1 Neural Network Surrogate Modeling Results

The neural network surrogate model was trained and tested as described previously with 60000 epochs with early stopping incorporated if no improvement of validation loss occurred within 1000 epochs. The results of one of the testing sets is shown in Figures 4.1 and 4.2 below.

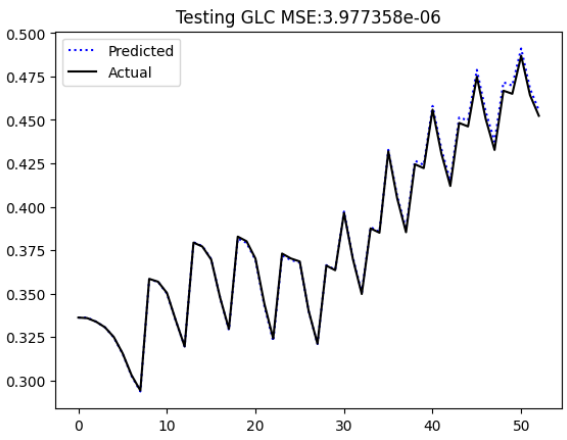

Figure 4.1: Neural network prediction for glucose level trajectory.

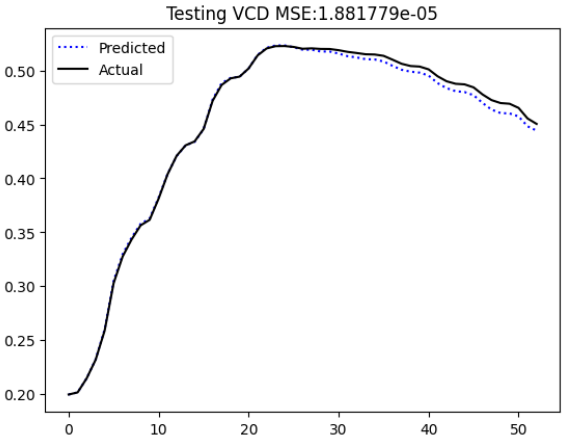

Figure 4.2: Neural network prediction for VCD trajectory.

The mean-squared-error (MSE) for the glucose and VCD output profiles were 3.977×10^{-6} and 1.882×10^{-5} respectively, and the surrogate model was considered satisfactory for use in the proposed method. The final structure and hyperparameters are shown in Table 4.1 below.

Table 4.1: Final structure of the neural network surrogate model.

Batches	5000
Hidden Layers	10
Nodes	500
Learning Rate Decay	0.99
Max Epochs	60000
Mini-batch size	52
Early Stop Threshold	1000
Training sets	4497
Validation sets	500
Testing sets	3
Final training loss	6.04e-07
Final validation loss	7.03e-07

4.2 Nested Sampling Results

The proposed method was tested against 3 scenarios. The first was using HSSM simulated data to showcase the method in the best case scenario where the model structure is identical to the 'process'. The second scenario applies the method on the actual experimental data to provide more realistic results. Normal distribution of the error is assumed as a base case for these two scenarios, with the standard deviation σ staying constant at 0.3. In the final scenario, the method is applied to the experimental data, and in addition, the likelihood function is adjusted to accommodate the realistic measurement noise of the viable cell density by reflecting the value's increase as cell density increases as described in Section 3.3.

For the first scenario, the algorithm was provided 8 batches of HSSM simulated data with a predefined maximum growth rate and primary death rate of 1.14 and 0.017, respectively to determine the technique's performance when the parameters are known. Nested sampling was able to produce the marginalized posterior probability of both parameters, as shown in Figure 4.3 below.

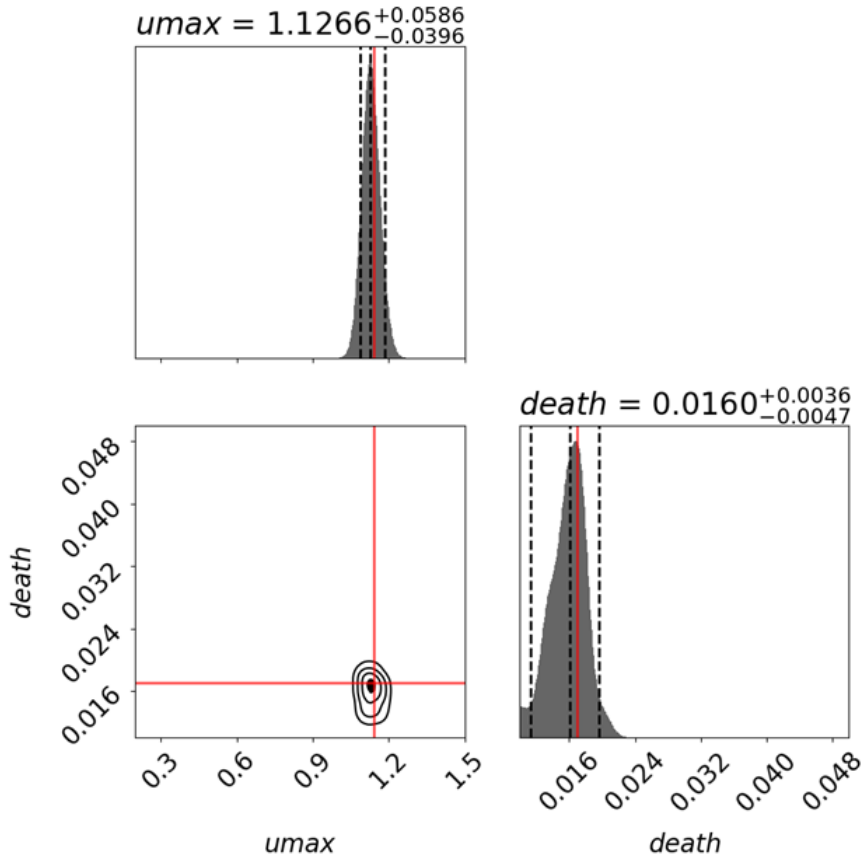

Figure 4.3: Corner plot showing the 1D and 2D marginalized posterior probability distribution of the parameters of interest. The dashed black lines are the 2.5%, 50% and 97.5% percentiles depicting the credible region. The red lines depict the "true" parameter values.

These plots show that the algorithm was able to estimate the parameters' "true" values within the credible region. To further validate the performance of the technique on simulated data, the median of the distribution of each parameter (ie. 1.1266, 0.016) was used in the neural network with 4 batches of unseen data and plotted against the same 4 batches with the "true" values. Results varied for different batches, but yet enabled broadly the capturing of the process dynamics as shown in Figure 4.4 below.

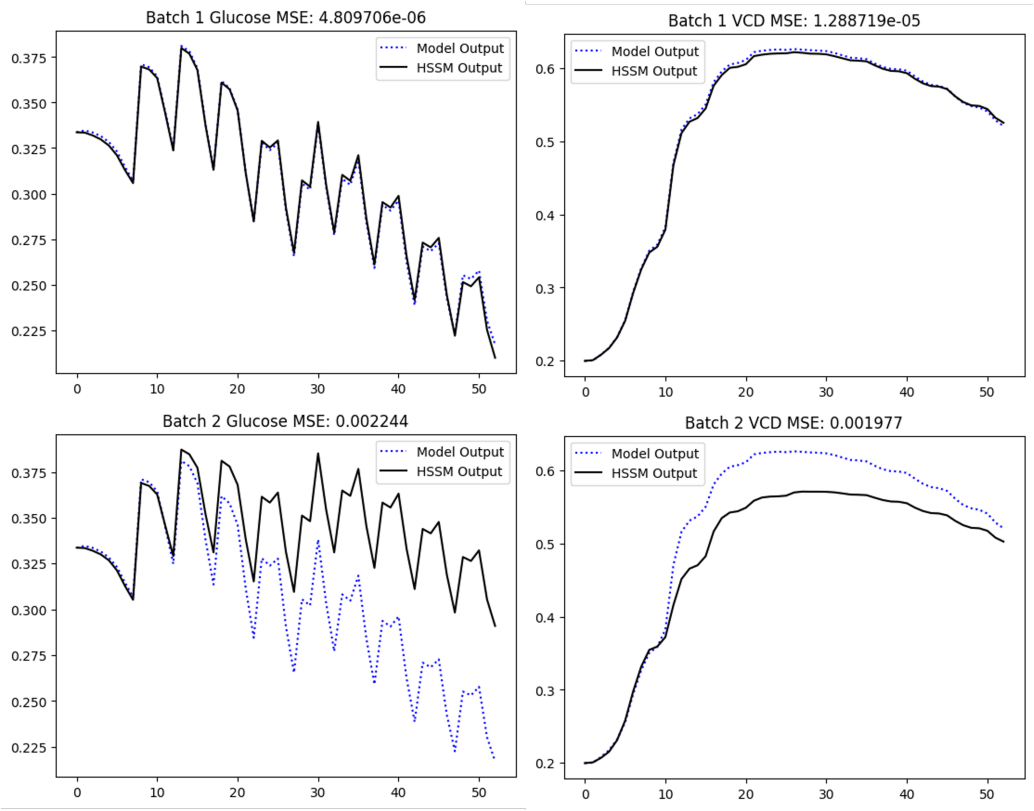

Figure 4.4: VCD and glucose level profile results of median estimated parameters applied to neural network for 2 batches of unseen data. Blue dotted line indicates the median values (1.1266, 0.016) were used. Black solid line indicates the "true" values (1.14, 0.017) were used.

The technique was then applied to the real experimental data. The algorithm was provided 8 of the 12 batches to estimate the parameters, leaving 4 batches for testing. The resulting corner plot is shown in Figure 4.5 below.

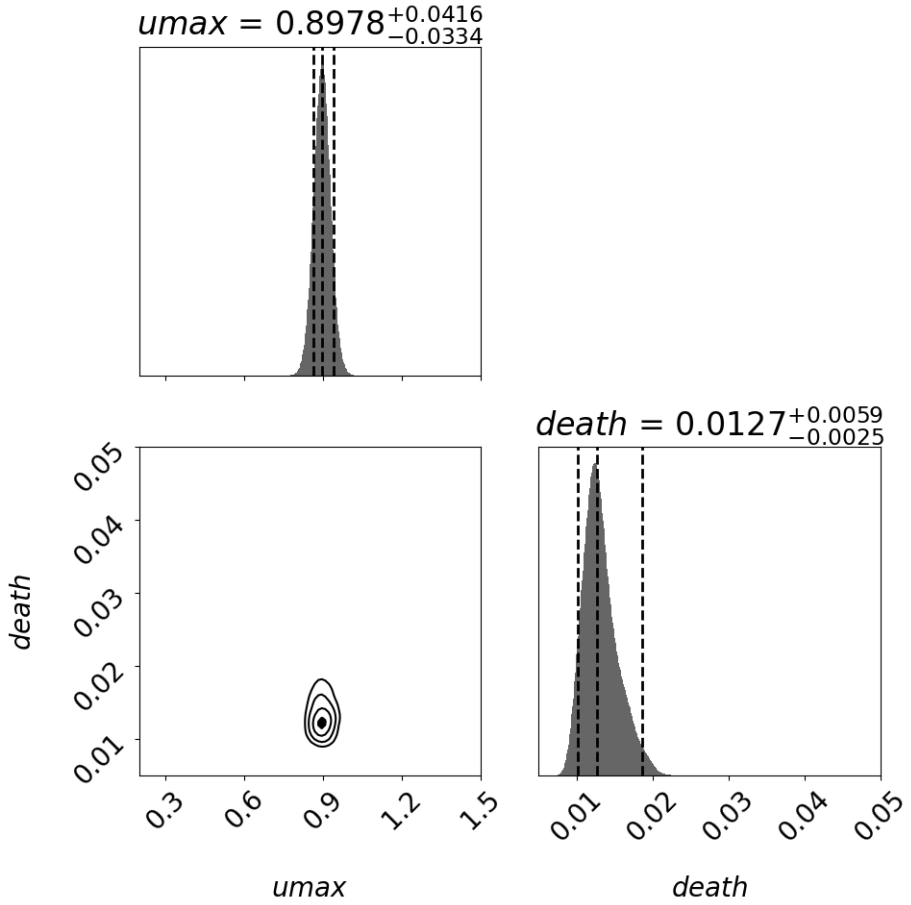

Figure 4.5: Corner plot showing the 1D and 2D marginalized posterior probability distribution of the parameters of interest for the experimental data. The dashed black lines are the 2.5%, 50% and 97.5% percentiles depicting the credible region.

The median parameter estimates (0.8978, 0.0127) were tested on the remaining batches 5, 7, 8 and 12. The resulting plots of batch 8 are shown in Figures 4.6 and 4.7 below. The HSSM results with these parameters are also shown to confirm the neural network is accurately replicating the HSSM. These results varied depending on the batch, in part due to the fact that the batch data showed significant variation, yet the estimated model parameters were able to replicate the behavior of the test data reasonably well.

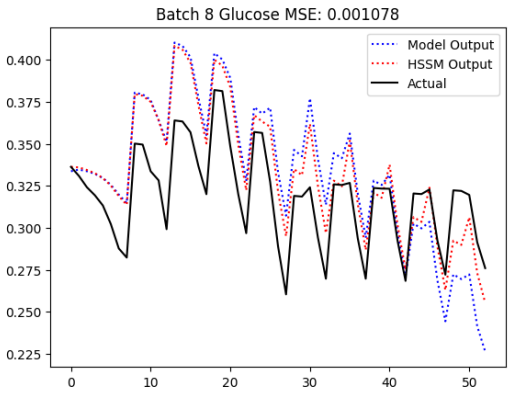

Figure 4.6: Glucose level trajectory for experimental data batch 8. Blue dotted line is the neural network prediction using median parameter estimates. Red dotted line is the HSSM prediction using median parameter estimates. Black solid line is the actual data.

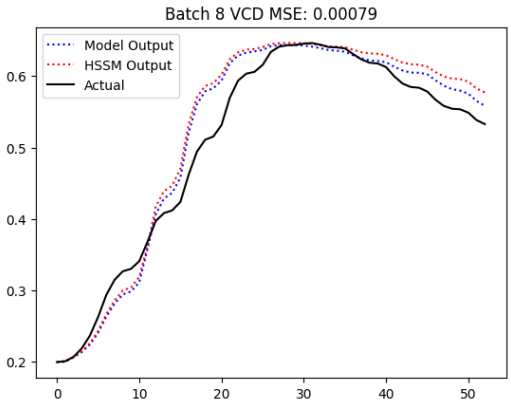

Figure 4.7: VCD trajectory for experimental data batch 8. Blue dotted line is the neural network prediction using median parameter estimates. Red dotted line is the HSSM prediction using median parameter estimates. Black solid line is the actual data.

The previous scenario was run again with a change in the standard deviation of the VCD. The results of this change are shown in Figures 4.8, 4.9, and 4.10 below. There was a slight increase in the MSE of the glucose output profile and a slight decrease in the MSE of the VCD output profile for batch 8.

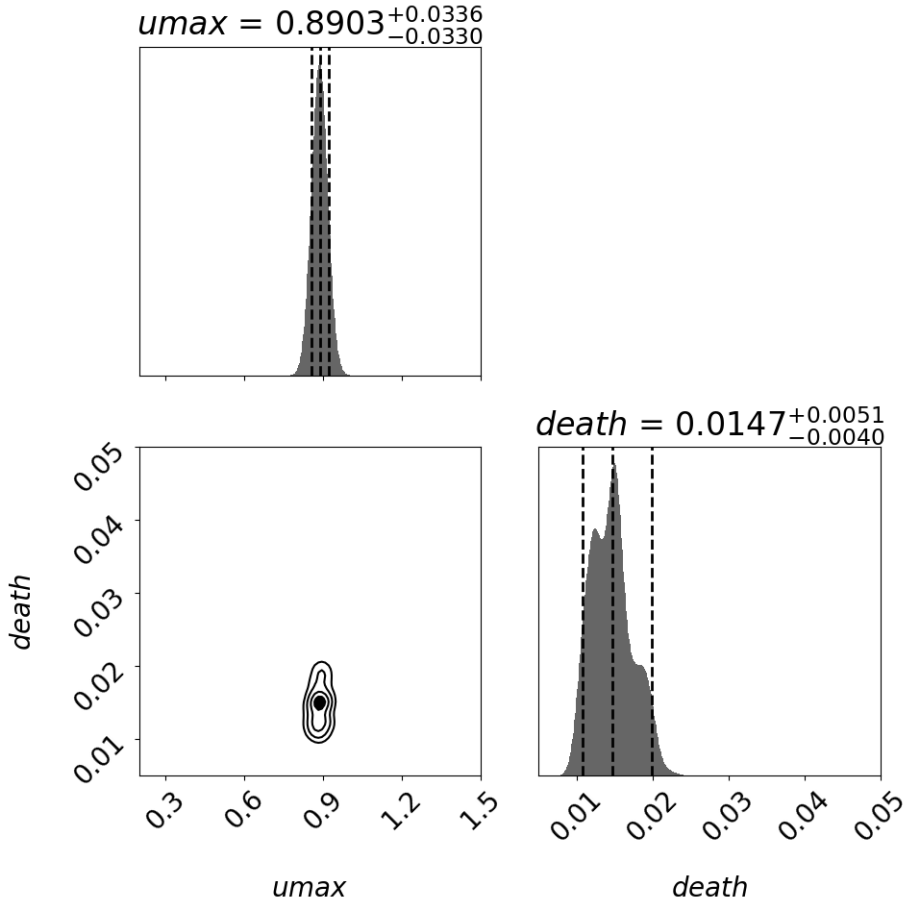

Figure 4.8: Corner plot showing the 1D and 2D marginalized posterior probability distribution of the parameters of interest for the experimental data when the standard deviation of the VCD was changed. The dashed black lines are the 2.5%, 50% and 97.5% percentiles depicting the credible region.

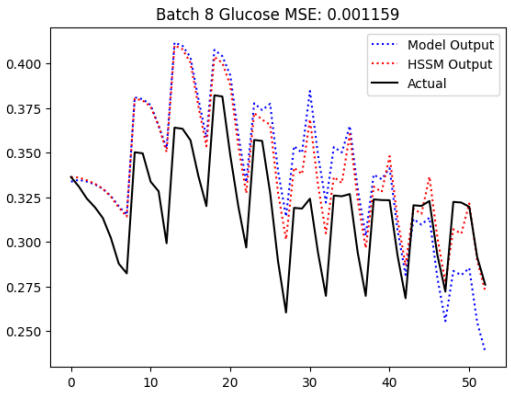

Figure 4.9: Glucose level trajectory for experimental data batch 8 with adjusted standard deviation. Blue dotted line is the neural network prediction using median parameter estimates. Red dotted line is the HSSM prediction using median parameter estimates. Black solid line is the actual data.

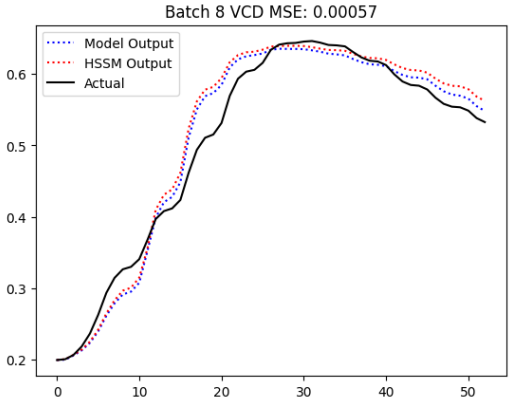

Figure 4.10: VCD trajectory for experimental data batch 8 with adjusted standard deviation. Blue dotted line is the neural network prediction using median parameter estimates. Red dotted line is the HSSM prediction using median parameter estimates. Black solid line is the actual data.

Finally, the nested sampling results were resampled such that their weights are equal in order to validate the algorithm results. This is shown using batch 8 in Figure 4.11 below. The lower blue line indicates the output predictions of the neural network when provided the parameter estimates that were calculated to be the 5th percentile of the nested sampling results. The upper blue line indicates the output predictions when provided the 95th percentile of the results. The results demonstrate the ability of the nested sampling algorithm to determine model parameters reasonably well using experimental data.

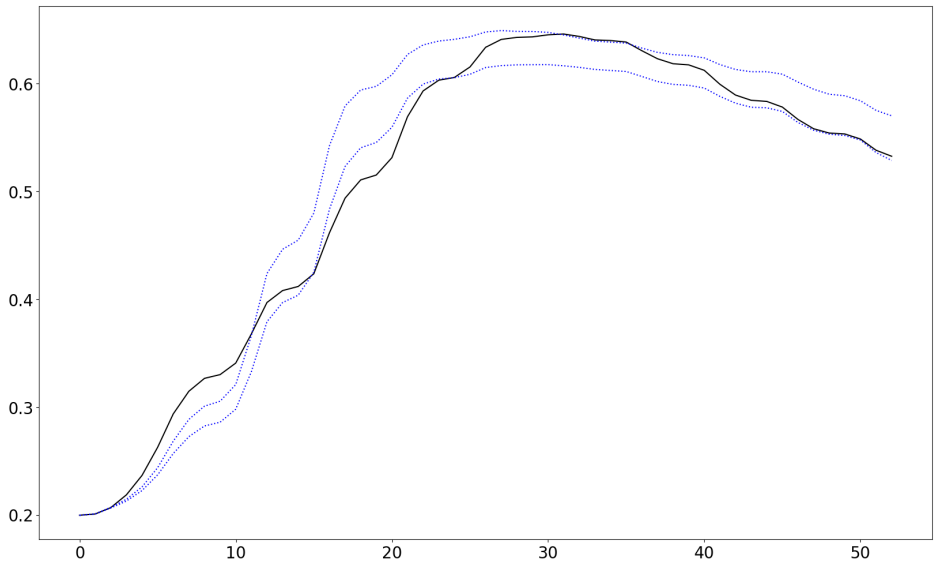

Figure 4.11: Plot of batch 8 VCD output profile with the neural network predictions when provided the 5% (bottom dotted blue line) and the 95% (top dotted blue line) percentiles of parameter estimates.

Chapter 5

Conclusions and Future Work

5.1 Conclusions

This thesis shows the development of a parameter estimation technique using a probabilistic approach that was demonstrated on experimental fed-batch bioreactor data. The method implemented Bayesian inference using Nested Sampling and determined optimal estimates via a likelihood function, which used model output from a neural network surrogate model to reduce computational load. The method was applied to model simulated data, real experimental data, and experimental data with varying measurement noise to demonstrate its feasibility, with favourable results.

5.2 Future Work

By using the method to calculate the parameter estimates along with a credible interval, this information can be used to guide future experiments. With the knowledge of what experimental factors generate the range of probable parameter values, future

experiments can be fine tuned based on previous parameter estimates and what is desired for the results. It is recommended that the measurement noise of the experimental data should be investigated further in order to achieve a more accurate distribution to provide to the likelihood function. This can be done, for example, by analyzing triplicates of measured data for each measurement device used in a conducted experiment.

Bibliography

[1] G. Ashton, N. Bernstein, J. Buchner, X. Chen, G. Csányi, A. Fowlie, F. Feroz, M. Griffiths, W. Handley, M. Habeck, et al. Nested sampling for physical scientists. *Nature Reviews Methods Primers*, 2(1):39, 2022.

[2] A. Botton, G. Barberi, and P. Facco. Data augmentation to support biopharmaceutical process development through digital models—a proof of concept. *Processes*, 10(9):1796, 2022.

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

[4] S. Cho, M. Kim, B. Lyu, and I. Moon. Optimization of an explosive waste incinerator via an artificial neural network surrogate model. *Chemical Engineering Journal*, 407:126659, 2021.

[5] S. Craven, N. Shirsat, J. Whelan, and B. Glennon. Process model comparison and transferability across bioreactor scales and modes of operation for a mammalian cell bioprocess. *Biotechnology progress*, 29(1):186–196, 2013.

[6] M. Elff. Estimation techniques: Ordinary least squares and maximum likelihood.

The Sage handbook of regression analysis and causal inference. Los Angeles: Sage Reference, pages 7–30, 2015.

[7] I. Fahmi and S. Cremaschi. Process synthesis of biodiesel production plant using artificial neural networks as the surrogate models. *Computers & Chemical Engineering*, 46:105–123, 2012.

[8] R. A. Fisher. On the mathematical foundations of theoretical statistics. *Philosophical transactions of the Royal Society of London. Series A, containing papers of a mathematical or physical character*, 222(594-604):309–368, 1922.

[9] 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*, 116(11):2944–2959, 2019.

[10] T. Hernández Rodríguez, C. Posch, R. Pörtner, and B. Frahm. Dynamic parameter estimation and prediction over consecutive scales, based on moving horizon estimation: applied to an industrial cell culture seed train. *Bioprocess and biosystems engineering*, 44(4):793–808, 2021.

[11] M. S. Hong, K. A. Severson, M. Jiang, A. E. Lu, J. C. Love, and R. D. Braatz. Challenges and opportunities in biopharmaceutical manufacturing control. *Computers & Chemical Engineering*, 110:106–114, 2018.

[12] K. Hornik, M. Stinchcombe, and H. White. Multilayer feedforward networks are universal approximators. *Neural networks*, 2(5):359–366, 1989.

[13] S. Jang and R. Gopaluni. Parameter estimation in nonlinear chemical and biological processes with unmeasured variables from small data sets. *Chemical engineering science*, 66(12):2774–2787, 2011.

[14] K. Jeon, S. Yang, D. Kang, J. Na, and W. B. Lee. Development of surrogate model using cfd and deep neural networks to optimize gas detector layout. *Korean Journal of Chemical Engineering*, 36:325–332, 2019.

[15] K.-R. Koch. *Introduction to Bayesian statistics*. Springer Science & Business Media, 2007.

[16] 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*, 170:107026, 2021.

[17] 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*, 167:34–44, 2022.

[18] J. S. Kwon, M. Nayhouse, and D. N. P. D. Christofides. Handling parametric drift in batch crystallization using predictive control with r2r model parameter estimation. *IFAC-PapersOnLine*, 48(8):912–917, 2015.

[19] Y. Liu and R. Gunawan. Bioprocess optimization under uncertainty using ensemble modeling. *Journal of biotechnology*, 244:34–44, 2017.

[20] L. Mesin. A neural algorithm for the non-uniform and adaptive sampling of biomedical data. *Computers in Biology and Medicine*, 71:223–230, 2016.

[21] H. Narayanan, M. F. Luna, M. von Stosch, M. N. Cruz Bournazou, G. Polotti, M. Morbidelli, A. Butté, and M. Sokolov. Bioprocessing in the digital age: the role of process models. *Biotechnology journal*, 15(1):1900172, 2020.

[22] L. Park, C. Park, C. Park, and T. Lee. Application of genetic algorithms to parameter estimation of bioprocesses. *Medical and biological engineering and computing*, 35:47–49, 1997.

[23] J. Pinto, M. Mestre, J. Ramos, R. S. Costa, G. Striedner, and R. Oliveira. A general deep hybrid model for bioreactor systems: Combining first principles with deep neural networks. *Computers & Chemical Engineering*, 165:107952, 2022.

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

[25] 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*, 117(5):1357–1366, 2020.

[26] I. Saraiva, A. Vande Wouwer, and A.-L. Hantson. Parameter identification of a dynamic model of cho cell cultures: an experimental case study. *Bioprocess and biosystems engineering*, 38:2231–2248, 2015.

[27] 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*, 6:100071, 2023.

[28] D. Seliştеanu, D. Șendrescu, V. Georgeanu, M. Roman, et al. Mammalian cell culture process for monoclonal antibody production: nonlinear modelling and parameter estimation. *BioMed research international*, 2015, 2015.

[29] H. Shi, Y. Zhang, H. Wu, S. Chang, K. Qian, M. Hasegawa-Johnson, and J. Zhao. Continuous cnn for nonuniform time series. In *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 3550–3554. IEEE, 2021.

[30] J. Skilling. Nested sampling for general Bayesian computation. *Bayesian Analysis*, 1(4):833 – 859, 2006. doi: 10.1214/06-BA127. URL `https://doi.org/10.1214/06-BA127`.

[31] D. Solle, B. Hitzmann, C. Herwig, M. Pereira Remelhe, S. Ulonska, L. Wuerth, A. Prata, and T. Steckenreiter. Between the poles of data-driven and mechanistic modeling for process operation. *Chemie Ingenieur Technik*, 89(5):542–561, 2017.

[32] J. S. Speagle. Dynesty: a dynamic nested sampling package for estimating bayesian posteriors and evidences. *Monthly Notices of the Royal Astronomical Society*, 493(3):3132–3158, 2020.

[33] M. Torzoni, A. Manzoni, and S. Mariani. A multi-fidelity surrogate model for structural health monitoring exploiting model order reduction and artificial neural networks. *Mechanical Systems and Signal Processing*, 197:110376, 2023.

[34] 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)*, pages 4899–4905. IEEE, 2018.

[35] A. Tulsyan, B. Huang, R. B. Gopaluni, and J. F. Forbes. On-line bayesian parameter estimation in general non-linear state-space models: A tutorial and new results. *arXiv preprint arXiv:1307.3490*, 2013.

[36] R. van de Schoot, S. Depaoli, R. King, B. Kramer, K. Märtens, M. G. Tadesse, M. Vannucci, A. Gelman, D. Veen, J. Willemsen, et al. Bayesian statistics and modelling. *Nature Reviews Methods Primers*, 1(1):1, 2021.

[37] N. Wulkow, R. Telgmann, K.-D. Hungenberg, C. Schütte, and M. Wulkow. Deterministic and stochastic parameter estimation for polymer reaction kinetics i: theory and simple examples. *Macromolecular Theory and Simulations*, 30(6): 2100017, 2021.

[38] J. Xie, B. Huang, and S. Dubljevic. Transfer learning for dynamic feature extraction using variational bayesian inference. *IEEE Transactions on Knowledge and Data Engineering*, 34(11):5524–5535, 2021.

[39] Z. Xing, N. Bishop, K. Leister, and Z. J. Li. Modeling kinetics of a large-scale fed-batch cho cell culture by markov chain monte carlo method. *Biotechnology progress*, 26(1):208–219, 2010.

[40] J. Yu and S. J. Qin. Multimode process monitoring with bayesian inference-based finite gaussian mixture models. *AIChE Journal*, 54(7):1811–1829, 2008.

[41] J. Zhang, Y. Zheng, and D. Qi. Deep spatio-temporal residual networks for city-wide crowd flows prediction. In *Proceedings of the AAAI conference on artificial intelligence*, volume 31, 2017.