Bayesian Parameter Estimation with Informative Priors for Nonlinear Systems

Matthew C. Coleman
Dept. of Chemical Engineering and Materials Science, University of California, One Shields Avenue, Davis, CA 95616

David E. Block
Dept. of Chemical Engineering and Materials Science and Dept. of Viticulture and Enology, University of California, One Shields Avenue, Davis, CA 95616

DOI 10.1002/aic.10667
Published online September 30, 2005 in Wiley InterScience (www.interscience.wiley.com).

The estimation of parameters for nonlinear process models is often accomplished through optimization routines that search for a global optimum with respect to a least squares or weighted least squares criterion. While such an approach is often reasonable, it fails to account for all of the information that is available from the data and practitioner. Here we focus on the inclusion of prior knowledge in the estimation of parameters in nonlinear dynamic systems. We use the Bayesian paradigm to define the probability distribution over process model parameters, called the Bayesian posterior. The quantities associated with this posterior distribution (e.g., credible regions, means, modes) are estimated via Markov Chain Monte Carlo (MCMC) integration. We first give a short introduction to Bayesian parameter estimation and the role of MCMC in evaluating arbitrary probability distributions. Bayesian parameter estimation (via MCMC) is then applied to three case studies. The first case study shows the basic methodology of assigning and evaluating a Bayesian posterior to a simple problem that consists of estimating the mean and variance of a sample. The second case study uses prior information that specifies a preference for a particular type of reaction mechanism over another for a simulated fermentation system; the inclusion of such prior information is shown to improve the estimated values of the model parameters in situations where data are sparse or noisy (compared to the more common weighted least squares approach). The third case study develops a hybrid semi-parametric neural network (NN) model to predict time-dependent observed state variables (cell and protein concentration) in Escherichia coli *fermentations. An integral step in the development of this hybrid model is parameter estimation of a nonlinear dynamic model. A hybrid model developed from the Bayesian parameter estimates is shown to outperform a hybrid model developed from the weighted least squares parameter estimates for predicting the final protein yield of a test set.* *AIChE J,* 52: 651–667, 2006
Keywords: Bayesian parameter estimation, Murkov chain Monte Carlo integration, nonlinear dynamic models, fermentation, E. coli

Introduction

The estimation of physical quantities from observed data is an integral part in the control and optimization of chemical and biochemical processes. The difficulties in performing such tasks arise from having complex physical phenomena where only a limited amount of data is available, which may also include high levels of noise. Estimation of parameters in models that describe the time dependent nature of chemical and biochemical processes is a common extension of this problem. The general problem can be stated as determining a vector of parameters $\boldsymbol{\phi}$ (e.g., kinetic rates of reaction, stoichiometric

Correspondence concerning this article should be addressed to D. E. Block at deblock@ucdavis.edu.

coefficients) after observing some data $\boldsymbol{D}$. The data will usually consist of observed state variables (i.e., product or reactant concentrations) at N discrete time points, $\boldsymbol{D}=[\boldsymbol{t}=\{t_1, t_2, \ldots, t_N\}, \tilde{\boldsymbol{Y}}=\{\tilde{\boldsymbol{y}}_1, \tilde{\boldsymbol{y}}_2, \ldots, \tilde{\boldsymbol{y}}_N\}]$, where $\tilde{\boldsymbol{y}}_i$ represents multiple responses observed at time t_i.

Parameter estimation of nonlinear process models is most often accomplished by nonlinear weighted least squares (WLS). Reviews of least squares and related methods can be found in vanBoekel,[1] Donaldson and Schnabel,[2] and Mendes and Kell,[3] among others.[4,5] While such methods provide adequate solutions for many problems, they do not effectively handle prior knowledge and may not be robust in the presence of sparse or noisy process data. Bayesian methods are well suited for the inclusion of prior knowledge into statistical analyses. Box and Draper[6] introduced Bayesian methods for estimating parameters of chemical reaction models from multiresponse data. Our approach shares the same fundamentals; however, we focus on combining prior information from a practitioner (e.g., the assumption that product inhibition is more likely than substrate inhibition in an enzymatic reaction) along with observed data to improve the estimated values of model parameters ($\boldsymbol{\phi}$). It is shown here that process models developed from the Bayesian methods outperform those produced from the more commonly used nonlinear WLS methods for data sets that are sparse and noisy.

The Bayesian methods used here define the joint probability distribution over three types of parameters: model parameters (e.g., kinetic rates of reaction), noise parameters (e.g., the standard deviation of noise in the measured state variables), and hyper-parameters. Here, hyper-parameters are used to interpret the prior knowledge specified by the practitioner.[7] This joint probability distribution is referred to as the Bayesian posterior and describes the probable values for all parameters. The posterior distribution is used to calculate all parameter estimates of interest (e.g., means, modes, credible intervals). A major difficulty in the Bayesian approach is evaluating the posterior density, which requires complicated integrals to be performed over arbitrary probability distributions. Oftentimes these integrals are analytically intractable and difficult to approximate. Markov Chain Monte Carlo (MCMC) sampling has been shown to be effective at handling many of these intractable integrals.[8-10] We prefer MCMC methods to evaluate Bayesian posterior densities because they require the fewest assumptions about the posterior; with MCMC we do not have to rely upon Gaussian or other fixed form distributions (e.g., Student's t-distribution) to approximate the posterior distribution. Chen et al.[11] discuss in more detail the advantages of Monte Carlo sampling in state and parameter estimation of nonlinear dynamic systems.

Bayesian approaches (evaluated via MCMC) have been used previously to estimate growth parameters of microbial systems.[12,13] While both approaches used data from literature along with their own experience with the microbial systems as their source of prior information, they do not include the use of hyper-parameters as discussed in this article in their analysis. The use of such hyper-parameters allows for more general types of prior information in estimating parameters. One example of this would be if we know there are two types of growth inhibition occurring in a fermentation (e.g., inhibition due to accumulation of product or lack of substrate[14,15]), but we believe that product inhibition has a stronger influence than inhibition from a lack of substrate. In the approach explained here, hyper-parameters are used to specify such preferences. Then Bayesian probability theory is used to determine parameter values that agree with our prior specifications (e.g., product inhibition is more likely) while still explaining the observed data. The use of hyper-parameters in the presented Bayesian analyses has been adapted from regularization techniques. Regularization techniques were first developed by Tikhonov[16] to solve ill-posed mathematical problems that require additional information to find a unique solution. Regularization techniques have been interpreted several times under the Bayesian framework to solve various types of problems.[17,18] Here we adapt regularization techniques into the Bayesian framework and generalize them for the estimation of parameters of nonlinear dynamic models.

The first section of this article reviews the basic components of Bayesian inference that are critical in the applications used in this article, along with an introduction to MCMC sampling. The second section of this article discusses three case studies. Case study one uses time independent single response data (product yields from three processes) to illustrate the basic implementation of MCMC methods to evaluate a Bayesian posterior probability distribution that includes one model parameter, one noise parameter, and one hyper-parameter. The second case study analyzes data from a simulated fermentation system where the Bayesian posterior includes eight model parameters, three noise parameters, and two hyper-parameters. The inclusion of prior information in the Bayesian framework is shown to improve estimated parameter values when compared to a WLS approach. Case study three uses experimental fermentation data that have been collected in our lab. A model that describes time dependent process characteristics (e.g., maximum specific growth rate) is developed; the Bayesian posterior includes seven model parameters, three noise parameters, and three hyper-parameters. Parameter values are estimated for 25 experiments that vary in different process conditions (such as pH, temperature, media ingredients). These parameter estimates are then used to train NN models that relate the time dependent process characteristics to the various process conditions. A different set of 10 fermentations is then used to test the empirical process model. A process model that used Bayesian parameter estimates is shown to predict the test set better than a model that used nonlinear WLS parameter estimates. Overall, the Bayesian methods seem to be most advantageous in situations where only sparse or noisy data are available.

Materials and Methods

Fermentation processing

Case study three uses a historical *Escherichia coli* database that was generated using a strain obtained from Drs. William Bentley and Govind Rao at the University of Maryland, College Park, and the University of Maryland, Baltimore County, respectively. This *E. coli* strain, JM 105 (*F' Δlac-pro thi strA endA sbcB15 hspR4 tra36 pro AB*$^+$ *lacI*^{q} *–ZΔM15*), bears the plasmid [pBAD-GFP::CAT].[19] The GFP and CAT reporters each possess a ribosome-binding site; however, both are under the control of the pBAD promoter of the *ara*BAD (arabinose operon). *E. coli* JM105 [pBAD-GFP::CAT] was induced for

Table 1. Process Inputs for Experimental *E. coli* Database in Case Study Three (These inputs are not Included in the ODE Model of Eq. 26)

Input Class	#	Input Name	Range	Description
Fermentation conditions	1	pH	6.70-7.30	Controlled at set point with acid/base feed
	2	Temperature (°C)	30-37	Controlled at set point with cooling jacket
	3	DO (% saturation)	20-40	Controlled at set point via agitation
Media variables	4	Yeast extract concentration (g/L)	5-15	In initial medium
	5	Tryptone concenteration (g/L)	15-30	In initial medium
	6	Percent arabinose (% wt/vol)	0.05-0.20	Used for induction; three times on an hourly basis
	7	Induction time (h)	2-4	Time for initial induction following initiation of feed
	8	Yeast extract source	Difco, Tastone, Amberex	Source of yeast extract
	9	Feed strategy (mL/min/g/L)	$\frac{1}{10}, \frac{1}{5}, \frac{2}{5}$	Glucose feed rate set at a ratio proportional to the cell density after reaches
	10	Antifoam	Before, after	Antifoam added before or after autoclaving fermentors
	11	Antibiotic	No, yes	Ampicillin added to fermentor
Inoculum conditions	12	Inoculation volume (mL)	100-400	Volume used to inoculate 4 L of medium
	13	Inoculation time (h)	6-12	Time that inoculum is allowed to grow prior to use

expression of green fluorescent protein (GFP) by the addition of appropriate amounts of arabinose.[20]

All fed-batch experiments were carried out in four BioFlo 3000 fermentors, each with a 5 L working volume (New Brunswick Scientific, NJ). All inocula were grown in an Innova 4000 shaker at 37°C and 200 rpm and were added to the fermentors by gravity. Media for the fermentations (4 L initially per fermentor) were prepared by combining yeast extract (Amberex 900, Universal Flavors, Juneau, WI; Tastone 900, Universal Flavors, Juneau, WI; or Difco Yeast Extract, Becton-Dickinson, Sparks, MD), tryptone (Fisher BioTech, NJ), and NaCl (40 g/L, Fisher Scientific, USA). Feed solutions containing 400 g/L D(+)-glucose (Sigma Chemical Company, St. Louis, MO) in deionized water were sterilized separately by autoclaving. The glucose feed was started when the broth reached an optical density (600 nm) reading of 2.0 (0.4 g/L) and was adjusted every two hours to a rate proportional to the optical density. Induction was achieved by injecting L-arabinose (Sigma Chemical Company, St. Louis, MO) solution, every hour for three hours, beginning either two or four hours after the start of glucose feed. A stock solution of ampicillin (100 mg/mL) was filter sterilized and stored at 4°C; appropriate amounts, based on 4 L media, were added according to the experimental design. The pH was controlled by the addition of 2 N sulfuric acid and 2 N sodium hydroxide. The dissolved oxygen (DO) level inside the fermentor vessel was controlled using agitation, with the minimum and maximum levels of agitation set at 200 and 1000 rpm, respectively. The experiments were designed in such a manner that no addition of pure oxygen was required. The foam level was maintained by the addition of appropriate amounts (approximately 1 mL) of Antifoam 289 (Sigma Chemical Company, St. Louis, MO). All fermentation inputs along with their operating ranges are shown in Table 1. Thirty-five unique combinations of these fermentation inputs (along with three repeated combinations) were performed. Additional experimental details can be found in Buck et al.[21] and Coleman et al.[22]

Computation

All computational algorithms were implemented using MATLAB® v6.0 (The Math Works, Inc., Natick, MA), and run on personal computers (3 GHz, 2 GB RAM). Several MATLAB® toolboxes were also utilized: the optimization toolbox (The Math Works, Inc., Natick, MA) and the Netlab toolbox (Neural Computing Research Group, Aston University, Birmingham, UK).

Theory

Bayesian inference

The fundamental difference between Bayesian and traditional statistical methods is their interpretation of probability. Classical methods, also known as frequentist methods, perceive probability as the long-run relative frequency of occurrence determined by the repetition of an event. A Bayesian perceives probability as a quantitative description of one's degree of belief in a given proposition.[23,24] This interpretation of probability better enables a practitioner to account for prior information in a statistical analysis, which is the primary reason it is used here. Many texts are available for detailed discussion of Bayesian ideas[7,25,26]; a brief outline is given below.

Bayesian inference begins with the Bayes Theorem, which is stated:

$$\Pr(\boldsymbol{\theta}|\boldsymbol{D}) = \frac{\Pr(\boldsymbol{D}|\boldsymbol{\theta})\Pr(\boldsymbol{\theta})}{\Pr(\boldsymbol{D})}. \quad (1)$$

While $\boldsymbol{\theta}$ may represent any arbitrary hypothesis, here it is exclusively used to represent a vector of parameters and $\boldsymbol{D}$ represents some data that have been collected. $\Pr(\boldsymbol{\theta}|\boldsymbol{D})$ reads as the probability of $\boldsymbol{\theta}$ given that $\boldsymbol{D}$ has been observed and is referred to as the Bayesian posterior probability distribution of $\boldsymbol{\theta}$. This posterior distribution is a joint probability distribution over all parameters and is used to make all decisions with respect to $\boldsymbol{\theta}$. The Bayesian posterior is comprised of three

components: the sampling distribution $\Pr(\boldsymbol{D}|\boldsymbol{\theta})$, prior $\Pr(\boldsymbol{\theta})$, and marginal likelihood $\Pr(\boldsymbol{D})$ $\Pr(\boldsymbol{D}|\boldsymbol{\theta})$ represents the probability of data set $\boldsymbol{D}$ occurring given that the parameters $\boldsymbol{\theta}$ are known; however, we do not know $\boldsymbol{\theta}$, we only know the data set. When $\boldsymbol{D}$ is known and $\boldsymbol{\theta}$ is unknown, then this term is referred to as the likelihood of the parameters given the data and denoted $L(\boldsymbol{\theta}|\boldsymbol{D})$. In this article the likelihood always relates the observed data to a physical model through the use of model and noise parameters. $\Pr(\boldsymbol{\theta})$ specifies the possible values of $\boldsymbol{\theta}$ before any data has been observed and is referred to as the prior probability of $\boldsymbol{\theta}$. For example, if a parameter is known to be positive, then the prior distribution specifies such information (e.g., $\Pr(\theta > 0) = 1$). The prior distribution always specifies the feasible values of the model and noise parameters that are used in the likelihood. $\Pr(\boldsymbol{D})$ is essentially a normalizing constant that assures that the posterior integrates to unity; this quantity is sometimes called the marginal likelihood, global likelihood, evidence, or Bayes Factor. It is not necessary to calculate the marginal likelihood in the work presented in this article; all of the relevant information is contained in the product of the likelihood and prior or otherwise stated:

$$\Pr(\boldsymbol{\theta}|\boldsymbol{D}) \propto L(\boldsymbol{\theta}|\boldsymbol{D})\Pr(\boldsymbol{\theta}). \tag{2}$$

To illustrate the formulation of these terms, suppose an arbitrary process has been run three times and each time the product yield was recorded, $\boldsymbol{D} = [\boldsymbol{g} = \{98, 100, 102\}]$, where g_i is the observed product yield of the i^{th} process run. Next, we would like to infer the expected product yield of future process runs if the same condition sets are used. We will assume that each product yield is independent from all others and distributed normally with an unknown mean (β) and standard deviation (σ). The sampling distribution for a single new data point given the values of β and σ is described by

$$\Pr(g_i|\beta, \sigma) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{(g_i - \beta)^2}{2\sigma^2}\right). \tag{3}$$

The sampling distribution, which describes the possible values of any three observations, is then

$$\Pr(\boldsymbol{D}|\beta, \sigma) = \prod_i^3 \Pr(g_i|\beta, \sigma). \tag{4}$$

Rearranging Eq. 4 and specifying that the data $\boldsymbol{D}$ remains constant while the parameters β and σ are varied results in the likelihood of the mean product yield and standard deviation

$$L(\beta, \sigma|\boldsymbol{D}) = \frac{1}{\sigma^3(2\pi)^{3/2}} \exp\left(-\frac{\sum_i^3 (g_i - \beta)^2}{2\sigma^2}\right). \tag{5}$$

This likelihood uses one model parameter (β) and one noise parameter (σ) to describe how the observed data was generated.

Next, prior distributions must be assigned to β and σ. Two commonly used priors when very little is known about a parameter are

$$\Pr(\beta) \propto 1 \tag{6}$$

and

$$\Pr(\log(\sigma)) \propto 1. \tag{7}$$

Both of these priors are used to specify that no information is known about either parameter before any data has been observed; hence, they are referred to as non-informative priors. The second type of prior (Eq. 7) was first derived by Jeffreys[27] for spread parameters such as σ. A prior distribution that expresses a parameter in terms of its logarithm is illustrated in Figure 1. While such non-informative priors can be useful in certain situations, there are many dangers and caveats in using them in a Bayesian analysis, as will be seen in the first case study. For more information on non-informative priors, see Box and Tiao.[28]

Use of more informative priors is generally the preferred approach. In the example above, we know that the mean product yield (β) must be equal to or greater than zero and there is likely to be an upper limit that is impossible to reach. A prior that specifies minimum and maximum attainable values is expressed as

$$\Pr(\beta) \propto \begin{cases} 1 & \text{if } \beta_{\min} \le \beta \le \beta_{\max} \\ 0 & \text{otherwise} \end{cases}, \tag{8}$$

which states that all values between $\beta_{\min}$ and $\beta_{\max}$ are equally as probable while all other values are impossible. Such a prior is analogous to upper and lower bound constraints in nonlinear optimization. A slightly more informative prior over β may be expressed as a half Gaussian,

$$\Pr(\beta) \propto \begin{cases} \dfrac{1}{\sigma_\beta} \exp\left(-\dfrac{\beta^2}{2\sigma_\beta^2}\right) & \text{if } \beta \ge 0 \\ 0 & \text{otherwise} \end{cases}, \tag{9}$$

where σ_β is a spread parameter that describes how large of a value you believe β could be; setting σ_β equal to $\beta_{\max}/3$ would specify that you were 99.7% sure that β is less than $\beta_{\max}$ and values near zero are more probable than those near $\beta_{\max}$. Both $\beta_{\max}$ and σ_β are referred to as hyper-parameters; hyper-parameters are here defined as those associated with prior distributions and are used to interpret the information that is available from a practitioner before any data has arrived. All other types of parameters used in this article (model and noise parameters) are associated with the likelihood. Just as with the noise parameter above (σ) when the value of a hyper-parameter is not known, a prior distribution must be assigned to it as well. For example, minimum and maximum values may be assigned to σ_β,

$$\Pr(\log(\sigma_\beta)) \propto \begin{cases} 1 & \text{if } \sigma_{\beta\,\min} \le \sigma_\beta \le \sigma_{\beta\,\max} \\ 0 & \text{otherwise} \end{cases}. \tag{10}$$

Each type of prior distribution used in this article is illustrated in Figure 1.

Therefore, taking the product of the likelihood (Eq. 5), the

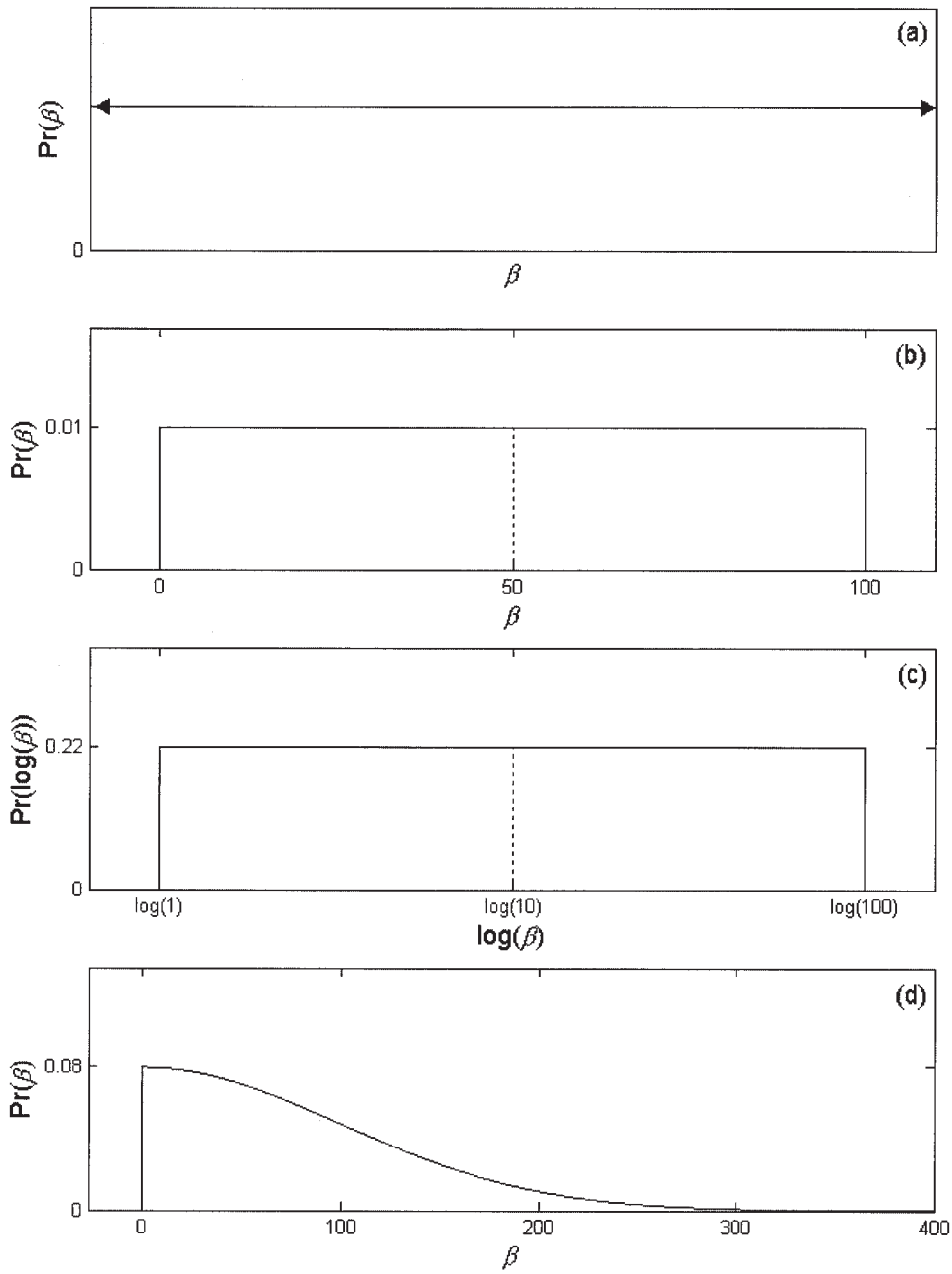

Figure 1. Four possible types of prior distributions:

(a) Non-informative prior defined over β, (b) flat prior with upper and lower bounds defined over β, (c) flat prior with upper and lower bounds defined over $\log(\beta)$, (d) half Gaussian prior defined over β with a mode of zero and standard deviation of 100. Distributions (b), (c), and (d) are all normalized so that the area under the distribution is equal to unity. Distribution (a) is referred to as an improper prior because it cannot be normalized. Distributions (b) and (c) are similar to each other; however, (b) specifies that a value of β between 0 and 50 is as probable as a value between 50 and 100, while (c) specifies that a value of β between 1 and 10 is as probable as a value between 10 and 100.

informative prior over β (Eq. 9), the non-informative prior over σ (Eq. 7), and the weakly informative prior over σ_β (Eq. 10) determines the form of the Bayesian posterior density for β, $\log(\sigma)$, and $\log(\sigma_\beta)$

$$\Pr(\boldsymbol{\theta}|\boldsymbol{D}) \propto L(\beta, \sigma|D)\Pr(\beta)\Pr(\log(\sigma))\Pr(\log(\sigma_\beta))$$

$$\propto \frac{1}{\sigma^3}\exp\left(-\frac{\sum_i^3 (g_i-\beta)^2}{2\sigma^2}\right)\frac{1}{\sigma_\beta}\exp\left(-\frac{\beta^2}{2\sigma_\beta^2}\right). \quad (11)$$

In this article $\boldsymbol{\theta}$ represents all of the parameters in the posterior (model, noise, and hyper) and for Eq. 11 $\boldsymbol{\theta} = [\beta, \log(\sigma), \log(\sigma_\beta)]$. Notice that the posterior is now defined over $\log(\sigma)$ and $\log(\sigma_\beta)$ and not σ and σ_β; this subtle difference is important but a discussion of it is outside the scope of this article.[28,29] The assignment of priors is a subjective matter and, therefore, Eq. 11 is only one possible Bayesian posterior.

The key points thus far are that the likelihood and priors are all that is required to define a Bayesian posterior. The likeli-

hood uses model and noise parameters to describe the physical phenomena being observed. The priors describe the probable values for all parameters before any data has been observed; prior distributions use hyper-parameters and range from being non-informative Eq. 6 or weakly informative Eq. 8 to informative Eq. 9. Choosing an appropriate prior is dependent upon what is known or believed about a particular parameter. The product of the likelihood and priors define the Bayesian posterior, which is the joint probability distribution for all parameters after data has been observed. Once a Bayesian posterior is defined, it must be evaluated, that is, we would like to determine the mode, mean, and credible intervals that are associated with each of the parameters. MCMC sampling is preferred for these purposes because fewer assumptions need to be made about the posterior than would be needed with other methods.

Markov Chain Monte Carlo methods

Calculating properties of the Bayesian posterior requires evaluating integrals of the form:

$$E[f(\boldsymbol{\theta})] = \int_{\boldsymbol{\theta}_{\min}}^{\boldsymbol{\theta}_{\max}} f(\boldsymbol{\theta}) \Pr(\boldsymbol{\theta}|\boldsymbol{D}) d\boldsymbol{\theta}. \tag{12}$$

If the probability distributions involved with such integrals have familiar fixed forms (i.e., Gaussian), then they may be analytically integrated. However, Bayesian posteriors most often have irregular forms that result in integrals that cannot be calculated analytically. In such situations it is required that a collection of points distributed according to the probability distribution be used to calculate the integrals. For example,

$$E[f(\boldsymbol{\theta})] = \frac{1}{N_S} \sum_{i}^{N_S} f(\boldsymbol{\theta}^{(i)}) \lim N_S \rightarrow \infty, \tag{13}$$

where ($\boldsymbol{\theta}^{(1)}$,. . . ,$\boldsymbol{\theta}^{(Ns)}$) are random samples determined by and drawn from the posterior distribution, can be used to calculate the integral of Eq. 12. This type of integration is known as Monte Carlo integration.[10,30] In complex situations, such as the analysis of nonlinear process models, independent values from the posterior cannot be directly sampled. However, because we can evaluate $\Pr(\boldsymbol{\theta}|\boldsymbol{D})$ for any value of $\boldsymbol{\theta}$, Markov Chains can be utilized to simulate dependent samples. A Markov Chain consists of N_S samples ($\boldsymbol{\theta}^{(1)}$,. . . $\boldsymbol{\theta}^{(i)}$,. . . $\boldsymbol{\theta}^{(Ns)}$), where the i^{th} sample is only dependent upon the preceding sample. For more information on Markov Chains, readers are referred to other sources.[10]

Our purpose for using Markov Chain Monte Carlo (MCMC) is then to draw samples from the posterior distribution and use these samples to estimate properties of the parameters in the posterior (i.e., means and credible intervals for $\boldsymbol{\theta}$). To do this we begin by defining an initial state $\boldsymbol{\theta}^{(1)}$ and then attempt to generate a new state (or sample) $\boldsymbol{\theta}^{(2)}$ that is distributed according to the posterior. We then must conceive a transition rule from $\boldsymbol{\theta}^{(i)}$ to $\boldsymbol{\theta}^{(i+1)}$ that will result in a Markov Chain that has the posterior distribution as its equilibrium distribution; that is, as the length of the chain approaches infinity, the distribution of the chain approaches $\Pr(\boldsymbol{\theta}|\boldsymbol{D})$. A simple algorithm was proposed by Metropolis et al.[31] that achieved this goal. A good review of the Metropolis algorithm and its generalizations can be found in Chib and Greenberg.[32] The result of an MCMC simulation is an $N_S \times N_P$ matrix of $\boldsymbol{\theta}$ samples,

$$\boldsymbol{\Theta} = \begin{bmatrix} \boldsymbol{\theta}_1^1 & \cdots & \boldsymbol{\theta}_{N_P}^1 \\ \vdots & \ddots & \vdots \\ \boldsymbol{\theta}_1^{N_S} & \cdots & \boldsymbol{\theta}_{N_P}^{N_S} \end{bmatrix}, \tag{14}$$

where there are N_P parameters in the posterior and N_S MCMC samples.

The most challenging issue with the Metropolis algorithm is choosing how to perturb $\boldsymbol{\theta}^{(i)}$ in calculating the next sample ($\boldsymbol{\theta}^*$). One approach that is straightforward to implement is to utilize a multivariate normal distribution to perturb $\boldsymbol{\theta}^{(i)}$. That is, we add some amount of noise to the current state of the chain ($\boldsymbol{\theta}^* = \boldsymbol{\theta}^{(i)} + \boldsymbol{\varepsilon}$), where $\boldsymbol{\varepsilon}$ is distributed normally (($\boldsymbol{\varepsilon} \sim N(0, \boldsymbol{\Sigma}_q)$)) and $\boldsymbol{\Sigma}_q$ is referred to as the covariance matrix of the proposal distribution. Roberts et al.[33] showed that if both the posterior distribution and proposal distributions are normal, then $\boldsymbol{\Sigma}_q$ should be adjusted so that the acceptance rate (frequency that $\boldsymbol{\theta}^*$ is accepted as the new state) is 0.45 for one dimensional distributions, 0.25 for distributions of six dimensions, and approach 0.23 as the number of dimensions approaches infinity. While most posteriors are not exactly normal, we use these as general guidelines to adjust $\boldsymbol{\Sigma}_q$.

The next important issue to deal with in MCMC simulation is to decide how many samples to collect, or when is it safe to assume the mean of the Markov Chain samples has converged to the mean of the posterior distribution. Collecting too few samples will result in inaccurate integration. A popular review of many approaches to diagnosing the convergence of Markov Chains can be found in Cowles and Carlin.[34] Here we adopt the use of potential scale reduction factors (($\hat{R}_j$)) to estimate when it is safe to stop the MCMC chain.[7] $\hat{R}_j$ is associated with the j^{th} parameter of $\boldsymbol{\theta}$ and estimates the potential improvement in the Markov Chain estimation of θ_j if the MCMC simulation were continued. At the end of an MCMC simulation, $\hat{R}_j$ values near one suggest that continuing the simulation would yield little improvement. In order to calculate $\hat{R}_j$, two or more parallel chains must be simulated. It is safe to stop an MCMC simulation when $\hat{R}_j$ values for all of the N_P parameters are close to one.[35]

In summary, a collection of points must be sampled from the posterior in order to find the means and credible intervals for each parameter. Markov Chain Monte Carlo methods are used to simulate samples from the Bayesian posterior. In the work presented here, we exclusively use a random walk Metropolis algorithm to perform Markov Chain simulation. Below, the use of these methods is illustrated in three case studies.

Results

Case study one: Estimation of the mean product yield

In a previous section we defined the form of a Bayesian posterior for the mean product yield (β), standard deviation (σ), and hyper-parameter (σ_β) of an arbitrary process (Eq. 11), where the hyper-parameter was used to specify a preference for smaller values of β. After some data have been observed ($\boldsymbol{D}$ =

[$\boldsymbol{g} = \{98, 100, 102\}$]), we would like to determine the values and credible intervals for each parameter. The main objective of this example is to examine how the informative prior over β (Eq. 9) affects its estimated value given various data sets with different noise levels. Suppose that previous experience with similar processes has shown mean product yields to be between 5 and 400. After a change in the process conditions, the mean product yield must be estimated from three experiments. We believe β is more likely to be closer to zero than 500 (the estimated upper limit of β); however, we are not sure what value to set σ_β (the spread parameter for the half Gaussian prior over β, which determines how much to shrink β). A prior over σ_β is then defined to explain our uncertainty in its value,

$$\Pr(\log(\sigma_\beta)) \propto \begin{cases} 1 & \text{if } 10 \le \sigma_\beta \le 500 \\ 0 & \text{otherwise} \end{cases}. \tag{15}$$

This prior over σ_β specifies a 68% certainty that β will always be less than 500, but we do not want to let the prior shrink values of β below 10. The posterior in this example is then the same as Eq. 11, where β is greater than or equal to zero and $\log(\sigma_\beta)$ has upper and lower bounds.

After establishing the form of the posterior, MCMC simulation is performed. To do this, we use the Metropolis algorithm (programmed in MATLAB® by the authors) to run three independent chains in parallel that use the same proposal covariance matrix $\boldsymbol{\Sigma}_q$. An iterative procedure is implemented to re-estimate $\boldsymbol{\Sigma}_q$ and adjust the acceptance rate to be between 0.2 and 0.3. This iterative procedure closely follows the suggestions of Gelman et al.[7] and continues until $\hat{R}_\beta$, $\hat{R}_{\log(\sigma)}$, and $\hat{R}_{\log(\sigma_\beta)}$ are all less than 1.1. The three chains are then combined to produce a $3N_S \times N_P$ matrix

$$\boldsymbol{\Theta} = \begin{bmatrix} \beta^{(1)} & \log(\sigma)^{(1)} & \log(\sigma_\beta)^{(1)} \\ \vdots & \vdots & \vdots \\ \beta^{(3N_S)} & \log(\sigma)^{(3N_S)} & \log(\sigma_\beta)^{(3N_S)} \end{bmatrix}. \tag{16}$$

Figures 2a, b, and c show the histograms for each column in matrix $\boldsymbol{\Theta}$ when data $\boldsymbol{D}^1$ has been observed (see Table 2). These histograms are discrete approximations to the marginal posteriors for each parameter. A marginal posterior is the one-dimensional probability distribution of a parameter where the full posterior is multidimensional. To evaluate the posterior in one dimension, we need to integrate with respect to all variables except one; for example,

$$\Pr(\beta|\boldsymbol{D}) = \int_{\log(10)}^{\log(500)} \int_{-\infty}^{+\infty} P(\beta, \log(\sigma), \log(\sigma_\beta)|\boldsymbol{D}) d\log(\sigma) d\log(\sigma_\beta), \tag{17}$$

where $P(\beta|\boldsymbol{D})$ is the marginal posterior of β. The first column of $\boldsymbol{\Theta}$ is a discrete approximation to $P(\beta|\boldsymbol{D})$; thus the expectation of β can be estimated by the mean of the first column. The marginal mode of β can be estimated by determining the highest peak in the histogram of Figure 2a. Credible intervals can similarly be estimated from the range of the histograms.

For comparison, Figure 3 also shows the marginal posteriors to β and $\log(\sigma)$ when a weakly informative prior is used for β, which results in

$$\Pr(\boldsymbol{\theta}|\boldsymbol{D}) \propto \frac{1}{\sigma^3} \exp\left(- \frac{\sum_i^3 (g_i - \beta)^2}{2\sigma^2} \right) \quad \text{if } 0 \le \beta \le 10^4 \tag{18}$$

as the full posterior. The marginal posteriors of Eq. 18 are available in closed form.[23] However, such closed form solutions are rarely (if ever) available for Bayesian posteriors associated with nonlinear dynamic processes.

Figure 2 shows how an informative prior over β influences the marginal posteriors given several additional data sets (shown in Table 2) that have varying levels of noise. Two types of prior distributions for β are compared in Figure 2, the half Gaussian informative prior (evaluated via MCMC) and the weakly informative prior (evaluated analytically). The mean value of the samples is identical for each data set ($\bar{g} = 100$), however, the standard deviation of the samples is different for each data set. It can be seen that the prior has very little influence upon the inference of β when the data strongly suggest that it is 100 (Figure 3a). However, the prior becomes more influential if the data weakly suggest that the mean product yield is 100 (Figure 3j). The histograms of Figure 2 use a weakly informative prior for the hyper-parameter σ_β (Eq. 15). Figure 3 shows the marginal posteriors for data sets $\boldsymbol{D}^1$ and $\boldsymbol{D}^4$ when a non-informative prior is used over the hyper-parameter, $\Pr(\sigma_\beta) \propto 1$. Notice that little has changed to the inference of β when $\boldsymbol{D}^1$ is observed; however, the marginal posteriors for $\boldsymbol{D}^4$ are drastically different. If the hyper-parameter is allowed to take very low values, the prior over β overwhelms the posterior and shrinks the estimated values of β to unreasonable values. Such phenomena are why it is preferable to use informative priors.

Case study two: Microbial growth inhibition

This second example deals with data that have been generated from a simulated batch fermentation system. We demonstrate that improved parameter estimates can be achieved by imposing informative priors on two of the model parameters. The model system is comprised of three state variables: cell concentration (X), substrate concentration (S), and product concentration (P). The system is defined by the following set of coupled ODEs

$$\begin{aligned} \frac{dX}{dt} &= \mu X \\ \frac{dS}{dt} &= -\frac{\mu X}{Y_{X/S}} \\ \frac{dP}{dt} &= Y_{P/X} \mu X \\ \mu &= \frac{\mu_{\max} S}{K_S + S} \left(1 - \frac{P}{K_p} \right) \end{aligned} \tag{19}$$

where $\mu_{\max}$ is the maximum growth rate, K_S is the Monod constant, K_P is a product inhibition term, $Y_{X/S}$ is the stoichiometric coefficient describing the formation of cell mass from substrate, and $Y_{P/X}$ is the stoichiometric coefficient describing

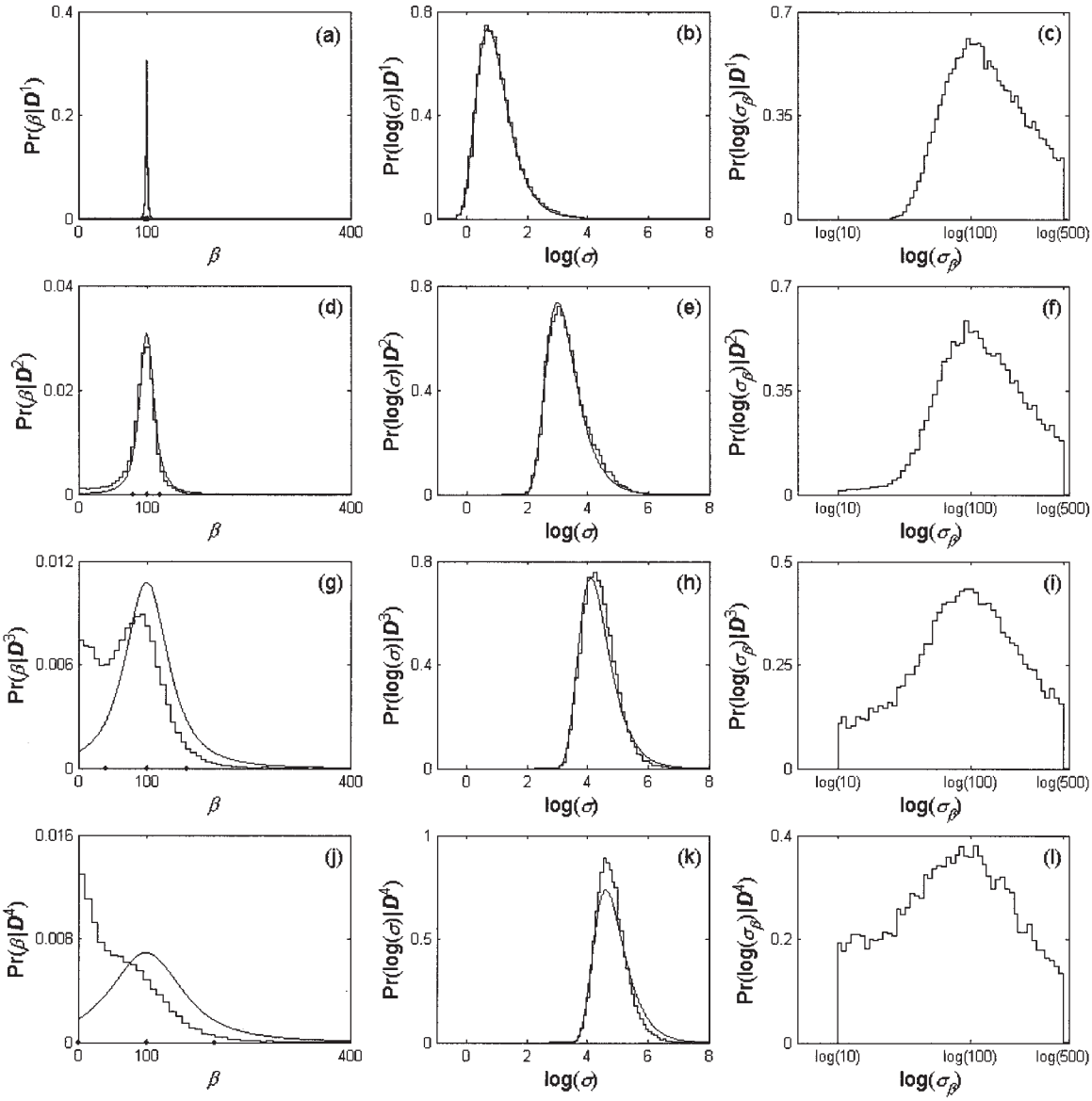

Figure 2. Approximated marginal posteriors of β, σ, and σ_β for the four different data sets D^1, D^2, D^3, and D^4 (see Table 2).

(a), (b), and (c) correspond to D^1. (d), (e), and (f) correspond to D^2. (g), (h), and (i) correspond to D^3. (j), (k), and (l) correspond to D^4. Histograms correspond to MCMC simulations of the Bayesian posterior with the informative prior over β (Eq. 11), and the smooth curves correspond to exact marginalization of the Bayesian posterior with all weakly informative priors (Eq. 18). The observed data points of each data set are also shown on the x-axis of (a), (d), (g), and (j).

the formation of product from cell mass. Observations of the state variables were simulated by solving the system of ODEs using MATLAB® function ODE23 (MATLAB®) and adding Gaussian noise,

$$\begin{aligned} y_{ji} &= f_j(\boldsymbol{\phi}, t_i) \\ \tilde{y}_{ji} &= y_{ji} + e_{ji} \\ e_{ji} &\sim N(0, y_{ji}/\alpha_j) \end{aligned} \tag{20}$$

where $y_{1i} = X_i = f_1(\boldsymbol{\phi}, t_i)$, $y_{2i} = S_i = f_2(\boldsymbol{\phi}, t_i)$, and $y_{3i} = P_i = f_3(\boldsymbol{\phi}, t_i)$ represent the state variables at time t_i, $\boldsymbol{\phi} = (X_0, S_0, P_0, \mu_{\max}, K_S, K_P, Y_{S/X}, Y_{P/X})$ is a vector of model parameters (including initial conditions), $\tilde{y}_{ji}$ represents the simulated observed j^{th} state variable where Gaussian noise is added to y_{ji}, e_{ji} is the noise added to the j^{th} state variable at time t_i, and α_j defines the signal to noise ratio for the j^{th} state variable. Parameter values used for simulation are shown in Table 3. Figure 4 shows a single simulated fermentation along with simulated observations taken every two hours.

The first approach taken to estimate the parameters was to perform weighted least squares optimization assuming that the true values of the noise parameters (α_j) shown in Table 3 are known. This assumption was made for simplicity, though in reality an independent estimate of experimental error would be sought. The minimized objective function was

Table 2. Utilized Data Sets in Case Study One*

Data Set	$\bar{g}$	Uniform Prior on β		Informative Prior on β	
		$E[\beta]$	95% CI	$E[\beta]$	95% CI
$\boldsymbol{D}^1$ 98, 100, 102	100	100.0	(95.0, 105)	100	(95.0, 104.8)
$\boldsymbol{D}^2$ 80, 100, 120	100	101.2	(56.1, 149.6)	94.7	(25.0, 138.6)
$\boldsymbol{D}^3$ 40, 100, 160	100	110.0	(0, 250.9)	78.7	(0.0, 154)
$\boldsymbol{D}^4$ 0, 100, 200	100	125.0	(0, 352.9)	73.8	(0.0, 182)

*Two different Bayesian posteriors were used to perform inference on the mean product yield (β). One posterior included a uniform prior over β, and the other included a half Gaussian prior over β.

$$WSSE = \sum_{j}^{3} \sum_{i}^{N} \frac{(\tilde{y}_{ji} - f_j(\boldsymbol{\phi}, t_i))^2}{\left(\frac{\tilde{y}_{ji}}{\alpha_j}\right)^2}, \tag{21}$$

where $\mathbf{D} = \{\tilde{\boldsymbol{y}}, \boldsymbol{t}\}$ represents the observed state variables at their respective times. The minimum and Hessian were estimated using the function LSQNONLIN (MATLAB® optimization toolbox) that included upper and lower bounds for each of the model parameters (see Table 3). These were then used to form a Gaussian approximation to the posterior.[2,4]

The second approach taken to estimate the parameters was to perform MCMC on a Bayesian posterior. The likelihood associated with the eight model parameters ($\boldsymbol{\phi}$) and three noise parameters ($\boldsymbol{\alpha}$) is

$$L(\boldsymbol{\phi}, \boldsymbol{\alpha}|\boldsymbol{D}) = \frac{1}{(2\pi)^{N/2} \prod_j^3 \prod_i^N \frac{\tilde{y}_{ji}}{\alpha_j}} \exp\left[-\frac{1}{2} \sum_{j}^{3} \sum_{i}^{N} \frac{(\tilde{y}_{ji} - f_j(\boldsymbol{\phi}, t_i))^2}{\left(\frac{\tilde{y}_{ji}}{\alpha_j}\right)^2} \right] \tag{22}$$

where the term in the exponent is equal to *WSSE*. Independent uniform priors over open intervals were assigned to most parameters. The bounds for each parameter are the same as those used for the WLS criterion and are shown in Table 3. More informative priors are assigned to the Monod constant K_S and the product inhibition term K_P. Suppose we believe that growth is more likely to be inhibited by the concentration of product rather than the lack of substrate. This type of informa-

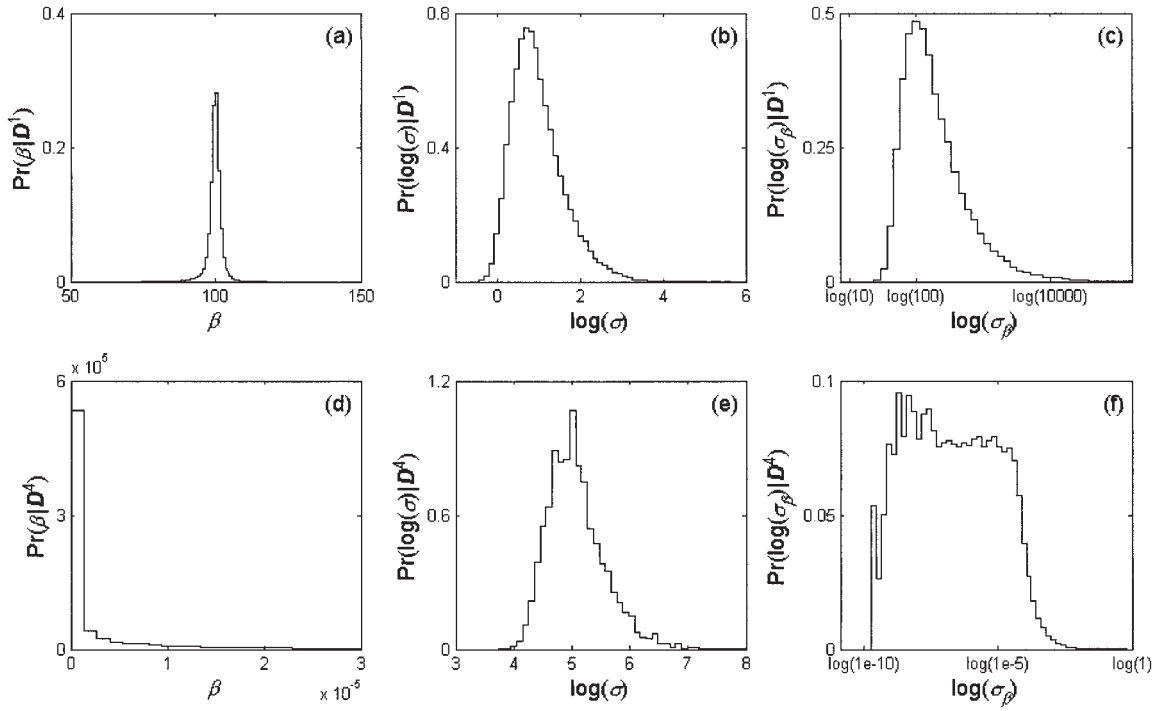

Figure 3. Approximated marginal posteriors of β, σ, and σ_β for data sets D^1 and D^4 (see Table 2).

(a), (b), and (c) correspond to D^1. (d), (e), and (f) correspond to D^4. A non-informative prior distribution over σ_β was used, $\Pr(\log(\sigma_\beta)) \propto 1$. Note that the x scales for D^4 distributions are different due to large differences in the predicted parameter values.

Table 3. List of Parameters for Case Study Two*

Parameter Type	Parameter	Lower Bound	Upper Bound	True Value
Model	Xo (g/L)	0	1	0.02
	So (g/L)	8	12	10
	Po (g/L)	0	1	0.01
	$\mu_{\max}$ (1/hr)	0	10	1
	K_S (g/L)	0	10	0.1
	K_P (g/L)	0	1000	50
	$Y_{X/S}$ (g/g)	0	100	.25
	$Y_{P/X}$ (g/g)	0	100	20
Noise	$\log(\alpha_X)$	log(1/1000)	log(1)	log(1/20) or (1/500)
	$\log(\alpha_S)$	log(1/1000)	log(1)	log(1/20) or (1/500)
	$\log(\alpha_P)$	log(1/1000)	log(1)	log(1/20) or (1/500)
Hyper	$\log(\sigma_{K_S})$	log(0.001)	log(10)	NA
	$\log(\sigma_{K_P})$	log(1)	log(1000)	NA

*Each parameter has upper and lower bounds that are used for both WLS and Bayesian approaches; however, in the Bayesian approach, the upper and lower bounds for K_S and K_P are replaced with half Gaussian priors defined by σ_{K_S} and σ_{K_D} (see Eq. 24). The true values are those used to simulate the data shown in Figure 4.

tion can be used to formulate prior distributions over K_S and K_P. By looking at the form of the ODE model above, we can see that this can be accomplished by specifying a preference for smaller values of K_S and K_P. Thus, we impose half Gaussian priors over K_S and K_P. For example,

$$\Pr(K_S) \propto \begin{cases} \dfrac{1}{\sigma_{K_S}} \exp\left(-\dfrac{K_S^2}{2\sigma_{K_S}^2}\right) & \text{if } K_S > 0 \\ 0 & \text{otherwise} \end{cases}, \tag{23}$$

where σ_{K_S} is a hyper-parameter that describes how large of a value you believe K_S may take. As seen in case study one, imposing this type of prior will shrink the estimated values of K_S and K_P in a manner that is consistent with the data. Upper and lower bounds are assigned to each hyper-parameter (see Table 3):

$$\Pr(\log(\sigma_{K_S})) \propto \begin{cases} 1 & \text{if } 0.001 < \sigma_{K_S} < 10 \\ 0 & \text{otherwise} \end{cases}. \tag{24}$$

This prior over $\log(\sigma_{K_S})$ specifies a range of priors that we are willing to consider for K_S. MCMC integration was performed as in case study one, but now the posterior included a total of 13 parameters (eight model, three noise, and two hyper).

These two approaches were then used to estimate the parameters given that only the four circled data points in Figure 4 have been observed. Figures 5a though 5e show the Gaussian approximations (via WLS) and histograms (via MCMC of the Bayesian posterior) to the marginal posterior of each model parameter (except initial conditions). Notice that both methods result in very similar distributions for $Y_{X/S}$ and $Y_{P/X}$. However, distributions for $\mu_{\max}$, K_S, and K_P are very different when the Bayesian methods are applied. The WLS estimate for K_S is 5.75; however, the data does not strongly suggest this value. The Bayesian approach shrinks the value of K_S to agree with both our prior knowledge and the data. A similar difference is seen in the Bayesian estimate of $\mu_{\max}$ because of correlations between $\mu_{\max}$, K_S, and K_P. Next, the two approaches were used to estimate the parameters when the signal to noise ratio for each parameter increased from α_j=20 to α_j=500. Figures 5f through i show the Gaussian approximations (via WLS) and histograms (via MCMC of the Bayesian posterior) to the marginal posterior of each model parameter (except initial conditions). As the signal to noise ratio increases, the estimated parameter values will converge upon the true parameter values.

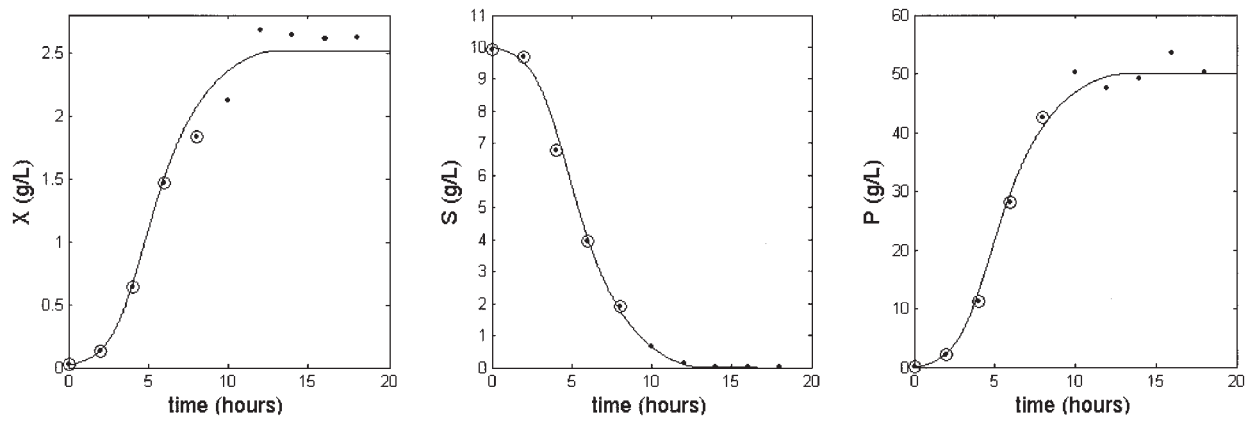

Figure 4. Simulated observations of cell (*X*), substrate (*S*), and product concentrations (*P*).

Smooth curves are generated from integrating the ODE model (Eq. 19) with the three parameter values specified in Table 3. Observations are simulated by adding Gaussian noise with a standard deviation of y_{ij}/α_j, where y_{ij} is the true value of the simulated j^{th} state variable at time t_i and α_j is the signal to noise ratio of the j^{th} state variable.

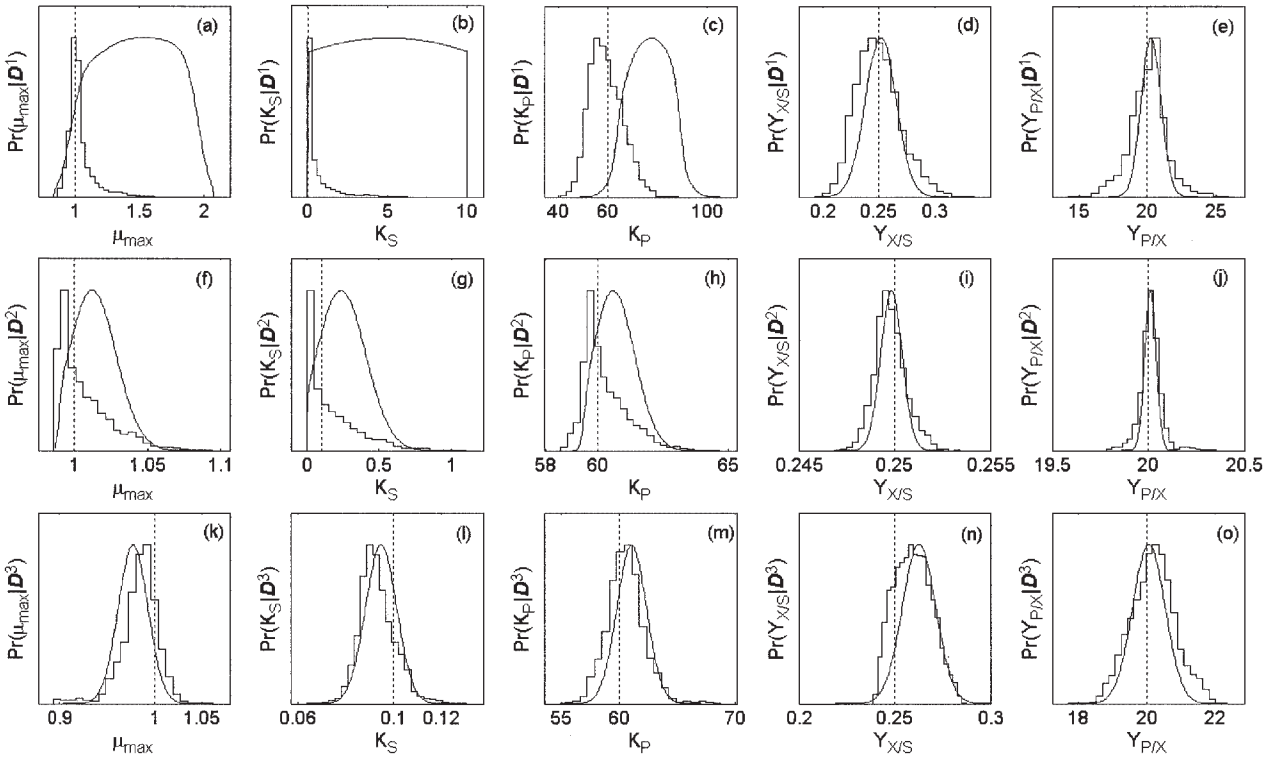

Figure 5. Approximated marginal posteriors of each model parameter for three different data sets, D^1, D^2 and D^3.

Sections (a) through (e) correspond to the first five points shown in Figure 4 (D^1). (f) through (j) correspond to the first five points shown in Figure 4 when noise level is reduced by a factor of 25 (D^2). (k) through (o) correspond to all ten points shown in Figure 4 (D^3). Notice how the Gaussian approximations of μ_{max}, K_S, and K_P for D^1 (a, b, and c) are all wide and weakly determined. Gaussian approximations were truncated for values that exceeded the upper and lower bounds (such as b), causing other parameter distributions to appear non-Gaussian (e.g., a). When the noise level is reduced (f, g, and h) or when additional data is observed (k, l, and m), the difference between the WLS Gaussian approximations and Bayesian histograms becomes less drastic. All vertical lines represent true parameter values used to simulate data observations.

Next, the two approaches were used to estimate the parameters given that all ten data points in Figure 4 have been observed. Figures 5k through o show the Gaussian approximations (via WLS) and histograms (via MCMC of the Bayesian posterior) to the marginal posterior of each model parameter (except initial conditions). Notice that all of the Gaussian approximations and histograms closely resemble each other. The additional data have overwhelmed the prior distributions because each parameter is now strongly determined by the data. Table 4 summarizes the estimated values for all parameters.

The main point in this example is that minimizing the error between model predictions and observed data does not always

Table 4. Estimated Values for Each Parameter Given Various Data Sets and Estimation Methods*

	D^1		D^2		D^3	
Parameters	WLS	Bayes	WLS	Bayes	WLS	Bayes
Xo	0.0201	0.0203	0.0200	0.0200	0.0199	0.0196
So	9.65	9.70	10.00	10.00	9.60	9.61
Po	0.0100	0.0101	0.0100	0.0100	0.0100	0.0101
μ_{max}	1.54	1.03	1.01	1.01	0.98	0.98
K_S	5.76	0.57	0.24	0.17	0.09	0.09
K_P	78.5	59.3	60.6	60.5	61.1	60.6
$Y_{X/S}$	0.25	0.25	0.25	0.25	0.26	0.26
$Y_{P/X}$	20.3	20.0	20.0	20.0	20.1	20.2
$\log(\alpha_X)$	NA	−2.69	NA	−6.02	NA	−2.68
$\log(\alpha_S)$	NA	−3.00	NA	−6.03	NA	−3.11
$\log(\alpha_P)$	NA	−2.52	NA	−5.80	NA	−2.85
$\log(\sigma_{K_S})$	NA	−1.30	NA	−1.89	NA	−1.96
$\log(\sigma_{K_P})$	NA	4.23	NA	4.31	NA	4.40
WSSE	2.69	3.03	9.80	10.48	6.54	8.06

*D^1 represents the first five data points (circled data points) shown in Figure 4. D^2 represents the same five data points when the observed noise values are reduced by a factor of 25. D^3 represents all ten data points shown in Figure 4. Noise and hyper parameters are not applicable (NA) for the WLS method (noise parameters are assumed to be known). The weighted sum of square error (*WSSE* Eq. 21) is also shown for each set of model parameters. WLS estimates are determined by the set of model parameters that minimized *WSSE.* Bayes estimates are the mean values of the Bayesian posterior.

yield optimal parameter estimates. Prior information about the value of a parameter can be used to improve estimating its value from data. While exact knowledge may not be available, general assumptions can be used to improve estimates in situations where data are sparse and noisy. As the amount of available data increases, prior information becomes less influential.

Case study three: Prediction of fermentation protein yield

In this third example, we analyze data from an experimental recombinant *E. coli* fermentation database generated in our lab. In this fermentation, *E. coli* produces green fluorescent protein (GFP). Two state variables were recorded over time (cell concentration and protein concentration), and fermentations were performed under various operating conditions (e.g., temperature, pH, media concentrations, amount of inoculum). The goal of this example is to build a model that predicts the time dependent nature of the state variables for novel combinations of operating conditions given a historical database.

The following model was proposed to describe the time dependent characteristics of the fermentations.

$$\begin{aligned}
\frac{dX}{dt} &= \mu \cdot X - \left(\frac{X}{V}\right)(F_I + F_S) \\
\frac{dV}{dt} &= F_I + F_S \\
\frac{dI}{dt} &= \left(\frac{C_I}{V}\right) \cdot F_I - q_I - \left(\frac{I}{V}\right)(F_I + F_S) \\
\frac{dP}{dt} &= \tau \cdot X - \left(\frac{P}{V}\right)(F_I + F_S) \\
\frac{dP_f}{dt} &= \left.\frac{dP}{dt}\right|_{t-t_{lag}} \\
\left.\begin{aligned}
\mu &= \mu_{\max} \\
\mu &= \mu_{\max}(X_L/X) \\
\mu &= \mu_{\max}(X_L/X)\exp(-K_D(X - X_D))
\end{aligned}\right\} &\begin{aligned} &\text{if } X \le X_L \\ &\text{if } X_L < X < X_D \\ &\text{if } X \ge X_D \end{aligned} \\
\left.\begin{aligned}
\tau &= 0 \\
\tau &= Y_{P/X} \cdot \mu
\end{aligned}\right\} &\begin{aligned} &\text{if } I = 0 \\ &\text{if } I \ge 0 \end{aligned}
\end{aligned} \tag{25}$$

The model consists of five state variables: (X) cell concentrations, (V) volume, (I) inducer concentration, (P) protein concentration, and (P_f) observable protein concentration; a time lag is associated with the production and detection of the GFP protein.[20] The model consists of seven parameters: $\boldsymbol{\phi} = (\mu_{\max}, X_L, X_D, K_D, Y_{P/X}, q_I, t_{lag})$, ($\mu_{max}$) maximum growth rate, ($X_L$) cell concentration where growth shifts from exponential to linear, (X_D) cell concentration where cells shift from linear growth to a death phase, (K_D) deceleration of growth rate in the death phase, ($Y_{P/X}$) stoichiometric coefficient describing the formation of protein from cell mass, (q_I) inducer (arabinose) consumption rate, and (t_{lag}) the lag time between protein production and detection. F_S is the substrate feed rate, and C_I is the known concentration of the inducer in the feed F_I. Figure 6 shows the observed cell and protein concentration data for a single fermentation along with simulations of the five state variables.

To model the relationship between the fermentation inputs (e.g., temperature and pH) and the state variables (cell concentration and protein concentration), a hybrid semiparametric neural network (NN) model approach is taken.[36] Here there are two components to the model (NN and ODE) that are connected in series. To test the accuracy of the developed hybrid model, ten of the 35 fermentations, each having a unique input condition set, were held out for a test set. The test set was randomly chosen; however, it was also verified to be representative of the entire range of final protein yields. The 25 remaining data sets were used to build a hybrid model that used the fermentation inputs to predict the state variables.

Two approaches were taken to estimate the values of $\boldsymbol{\phi}$ and $\boldsymbol{\alpha}$. First, traditional weighted least squares minimization using LSQNONLIN (MATLAB® optimization toolbox) was performed upon the same objective function shown in Eq. 21, with the exception that there were now only two state variables. $\boldsymbol{\alpha}$ was estimated after the *WSSE* (Eq. 21) was minimized with respect to $\boldsymbol{\phi}$. $\boldsymbol{\alpha}$ and $\boldsymbol{\phi}$ were iteratively re-estimated until there was no further change in $\boldsymbol{\phi}$ or $\boldsymbol{\alpha}$. Upper and lower bounds were also applied to each parameter (shown in Table 5). The estimated Hessian was also used to form a Gaussian approximation to each of the model parameters.

The second approach to estimate $\boldsymbol{\phi}$ and $\boldsymbol{\alpha}$ used MCMC to evaluate a Bayesian posterior. The likelihood of the parameters associated with a fermentation was then

$$L(\boldsymbol{\phi}, \boldsymbol{\alpha}|\boldsymbol{D}_{(k)}) \propto \frac{1}{\Pi_j^2 \Pi_i^N \frac{\tilde{y}_{ji}}{\alpha_j}} \exp\left[-\frac{1}{2}\sum_j^2 \sum_i^N \frac{(\tilde{y}_{ji} - f_j(\boldsymbol{\phi}, t_i))^2}{\left(\frac{\tilde{y}_{ji}}{\alpha_j}\right)^2}\right] \tag{26}$$

where $\boldsymbol{D}_{(k)}$ represents recorded state variables of the k^{th} fermentation, $\tilde{y}_{ji}$ is the observed value of the j^{th} state variable recorded at time t_i, $\boldsymbol{\phi}$ is a vector of the seven model parameters, $f_j(\boldsymbol{\phi},t_i)$ is the estimated value of the j^{th} state variable recorded at time t_i (solved by numerically integrating the ODE model for fixed values of $\boldsymbol{\phi}$ using ODE23 (MATLAB®)), and α_j is the signal to noise ratio of the j^{th} state variable. Initial concentrations for all state variables are assumed known. Uniform priors with upper and lower bounds (Table 5) were assigned to each parameter with the exception of K_D, q_I, and t_{lag}. Half Gaussian priors were assigned to K_D, q_I, and t_{lag}. Upper and lower bounds were assigned to each hyper-parameter describing the half Gaussian priors (Table 5). MCMC was performed on each data set as explained in case study one, but the posterior now includes seven model parameters ($\boldsymbol{\phi}$), two noise parameters ($\boldsymbol{\alpha}$), and three hyper-parameters ((σ_{K_D}, σ_{q_I}, *and* $\sigma_{t_{lag}}$). This resulted in 28 matrices $\boldsymbol{\Theta}_k$ corresponding to the 25 unique input fermentations in the training data set. The additional three data sets are from repeated fermentation input runs.

Table 6 summarizes the estimated parameters for each of the repeated fermentation data sets. Ideally if the same input conditions are repeated twice, the same parameter estimates should be reached; this is rarely the case for all parameters except for $\mu_{\max}$. By imposing the informative priors over K_D, q_I, and t_{lag}, more consistent parameter estimates are reached for different data sets performed with the same conditions. Notice that the informative priors over K_D, q_I, and t_{lag} have varying levels of influence in shrinking their estimated values. For example, the t_{lag} estimates for condition set 3 differ very little between WLS and Bayes. This suggests that the data from $\boldsymbol{D}_3^1$ and $\boldsymbol{D}_3^2$

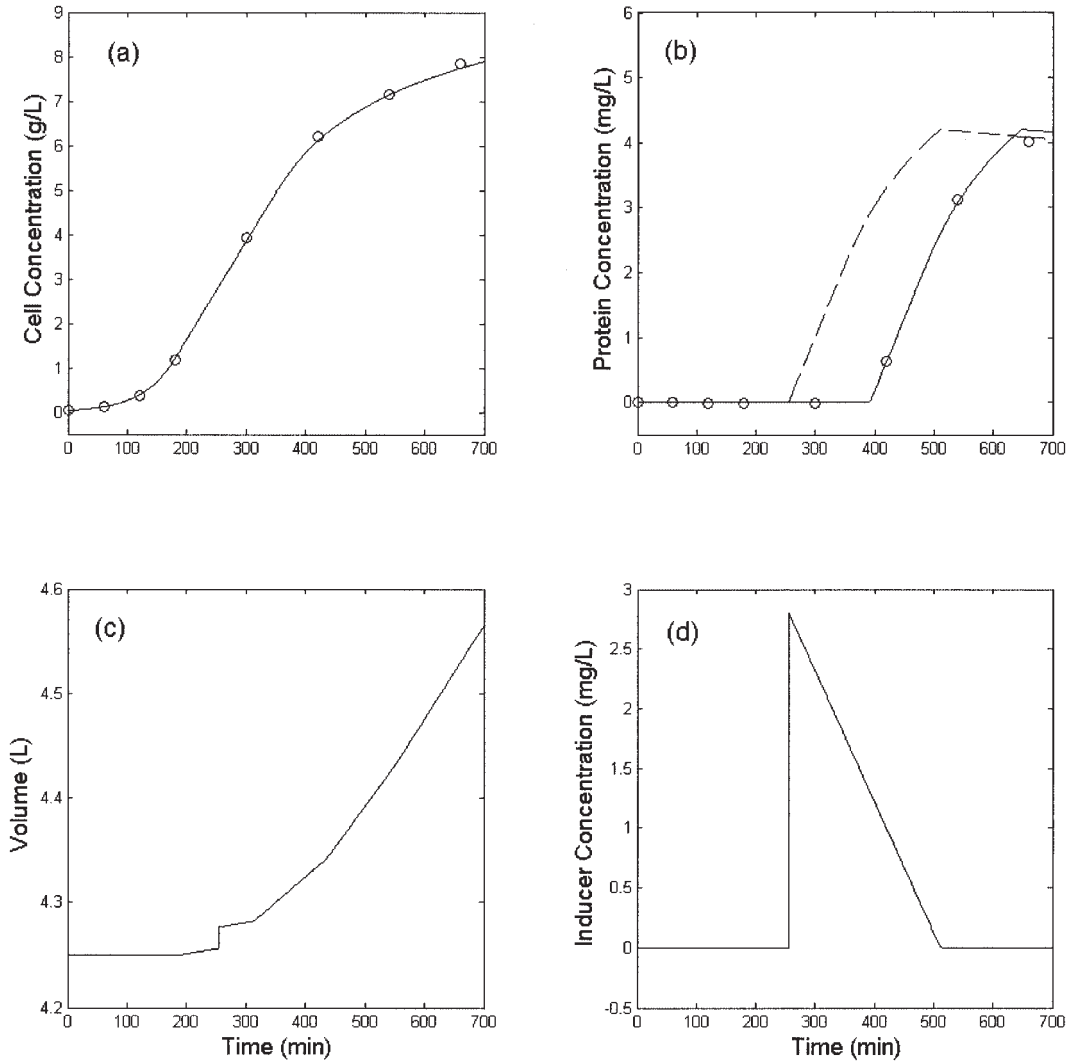

Figure 6. Observed cell and protein concentration data along with simulations of the five state variables for a typical *E. coli* fermentation.

All simulated data used parameters that were estimated by the approximated mode of the posterior that included all uniform priors. (a) cell concentration (X), (b) actual and observed protein concentration (P dashed, P_f solid), (c) volume (V), and (d) inducer concentration (I).

Table 5. Model, Noise, and Hyper Parameters Used in the Bayesian Posterior for the ODE Model (Eq. 25) Describing the Time Dependent Nature of the Experimental *E. coli* Fermentations (Each Parameter is Also Assigned Upper and Lower Bounds)

Parameter Type	Parameter	Lower Bound	Upper Bound
Model	$\mu_{\max}$ (1/min)	0	.5
	X_L (g/L)	0	10
	X_D (g/L)	2	20
	K_D (L/g)	0	10
	$Y_{P/X}$ (mg/g)	0	10
	q_I (mg/L/min)	10^{-4}	.5
	t_{lag} (min)	10	1000
Noise	$\log(1/\alpha_X)$	log(1)	log(1/100)
	$\log(1/\alpha_P)$	log(1)	log(1/100)
Hyper	$\log(\sigma_{KD})$	.2	5
	$\log(\sigma_{qI})$	10^{-3}	.5
	$\log(\sigma_{tlag})$	10	120

strongly suggest that t_{lag} is approximately 115 (min). There is a much larger difference between WLS estimates of t_{lag} for condition set 2 (25.6 and 68.1). When the Bayesian analysis is applied, these two estimates become much closer together (38.7 and 34.2). The increase from 25.6 to 38.7 is due to the integration of the posterior and not the implementation of the informative prior. However, the decrease from 68.1 to 34.2 is due to the informative prior shrinking the estimate. While we cannot show that the Bayesian estimates are closer to the true parameter values for this real case, we can show that they produce more consistent results between repeated fermentation runs. Each condition set had two repeated experiments ($\boldsymbol{D}_k^1$ and $\boldsymbol{D}_k^2$); however, only $\boldsymbol{D}_k^1$ from each of the condition sets was included in the NN training set.

Next, the NN model relating the fermentation inputs to the model parameters $\boldsymbol{\phi}$ was trained. To do this we utilized automatic relevance determination (ARD) with Gaussian Process NNs.[29,37] Each of the 13 process inputs was used as inputs to an

Table 6. Bayesian and WLS Estimated Values for the Model Parameters Corresponding to Repeated Fermentation Input Conditions*

	Condition Set 1				Condition Set 2				Condition Set 3			
	WLS		Bayes		WLS		Bayes		WLS		Bayes	
Parameter	D_1^1	D_1^2	D_1^1	D_1^2	D_2^1	D_2^2	D_2^1	D_2^2	D_3^1	D_3^2	D_3^1	D_3^2
μ_{max}	0.015	0.015	0.015	0.015	0.015	0.015	0.015	0.015	0.012	0.012	0.012	0.012
X_L	2.43	2.56	2.77	3.01	1.64	1.59	1.83	1.58	1.85	2.09	1.79	2.16
X_D	9.56	11.16	7.23	7.38	6.68	7.05	5.22	5.76	6.30	8.28	6.74	6.97
K_D	0.46	1.61	0.29	0.32	4.12	4.76	1.08	1.09	0.30	1.06	0.16	0.36
$Y_{P/X}$	4.45	3.58	2.93	2.93	3.62	3.41	3.77	3.68	0.16	0.17	0.17	0.17
q_I	0.064	0.060	0.026	0.033	0.014	0.008	0.015	0.016	0.024	0.010	0.009	0.007
t_{lag}	73.3	61.6	37.0	36.3	25.6	68.1	38.7	34.2	113.0	117.0	111.5	116.6

*For example, condition set 1 corresponds to a unique combination of fermentation inputs, and two fermentations were performed with these same exact fermentation conditions; D_1^1 then corresponds to the data collected during the first fermentation with condition set 1 and D_1^2 corresponds to the data collected during the second fermentation with condition set 1. Ideally D_k^1 and D_k^2 for each input combination will yield the same estimates of model parameters; however, due to the sparse and noisy nature of the collected data, estimated values differ.

NN that had one of the model parameters as its output. A 0.632 bootstrap was used to estimate the correlation coefficient between NN predictions and target model parameter values.[38] Table 7 summarizes the model fits for the training and test sets for each parameter. The trained NNs were then used to estimate the model parameters for the test set fermentations.

The trained NN and ODE models were then combined and used to predict the time dependent nature of the test set fermentations and, thus, estimate the final protein yield. Figure 7 shows the predictions of the final protein yields for the test set; 7a are the predictions when WLS estimates are used as targets for the NNs, and 7b are the predictions when the means of the Bayesian posteriors are used as the targets to the NNs. It can be seen that the Bayesian estimates significantly improved our ability to predict the protein yield of the test set.

By looking more closely at the NNs that predicted the model parameter $Y_{P/X}$, we can gain some insight as to why the Bayesian approach outperformed the WLS approach. Both NNs for $Y_{P/X}$ (WLS and Bayes) identified feed strategy as the most important process condition in predicting $Y_{P/X}$. Figure 8 plots the estimated $Y_{P/X}$ values versus the feed strategy; feed strategy here is the rate of substrate addition (ml/min) per current cell concentration (g/L), which is updated incrementally throughout the fermentation. It can be seen here that 8a suggests a feed strategy of 1/10 is probably optimal; however, setting the feed strategy at 2/5 could also potentially optimize $Y_{P/X}$. The Bayes estimates, shown in 8b, seem to suggest that $Y_{P/X}$ will definitely be optimized by setting feed strategy to 1/10. The data points

Table 7. Regression Coefficients for Each NN Used to Predict the Model Parameters (that is μ_{max}, X_L) Using the Fermentation Inputs (i.e., pH, temperature)*

	WLS		Bayesian	
	R_{train}	R_{test}	R_{train}	R_{test}
μ_{max}	0.69	0.80	0.59	0.83
X_L	0.37	0.60	0.49	0.74
X_D	0.38	0.72	0.40	0.65
K_D	0.35	0.32	0.29	0.47
$Y_{P/X}$	0.47	0.71	0.68	0.88
q_I	0.43	0.46	0.51	0.75
t_{lag}	0.36	0.12	0.40	0.50

*R_{train} is a .632 bootstrap estimate with 50 iterations using the 25 training points. R_{test} is a hold out estimate using the 10 test points. In general, the regression coefficients are higher when Bayesian estimates are used as NN targets.

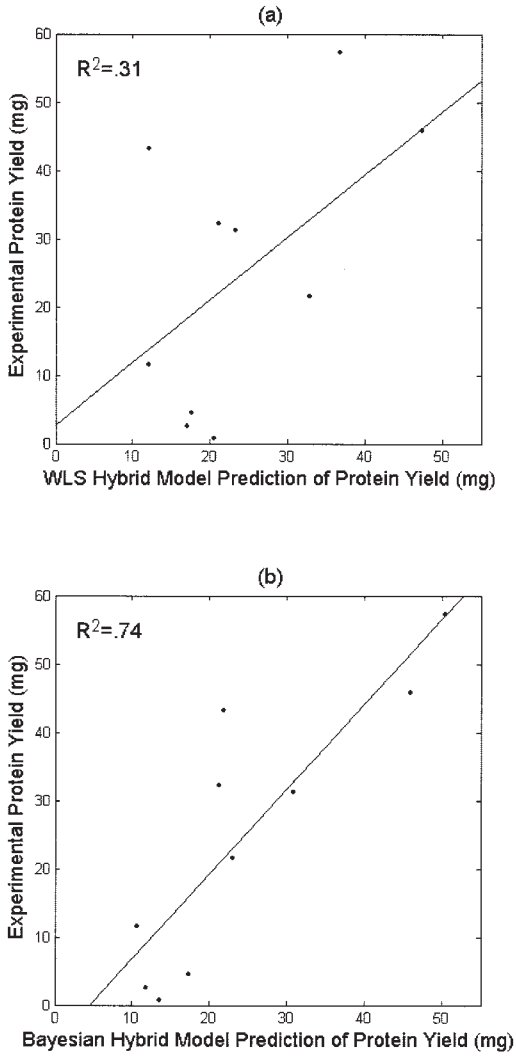

Figure 7. Actual vs. predicted protein yields.

(a) Corresponds to predictions made by a hybrid-NN when WLS estimates of model parameters are used to train the NN component of the hybrid model. (b) corresponds to predictions made by a hybrid-NN when Bayesian estimates of model parameters are used to train the NN component of the hybrid model.

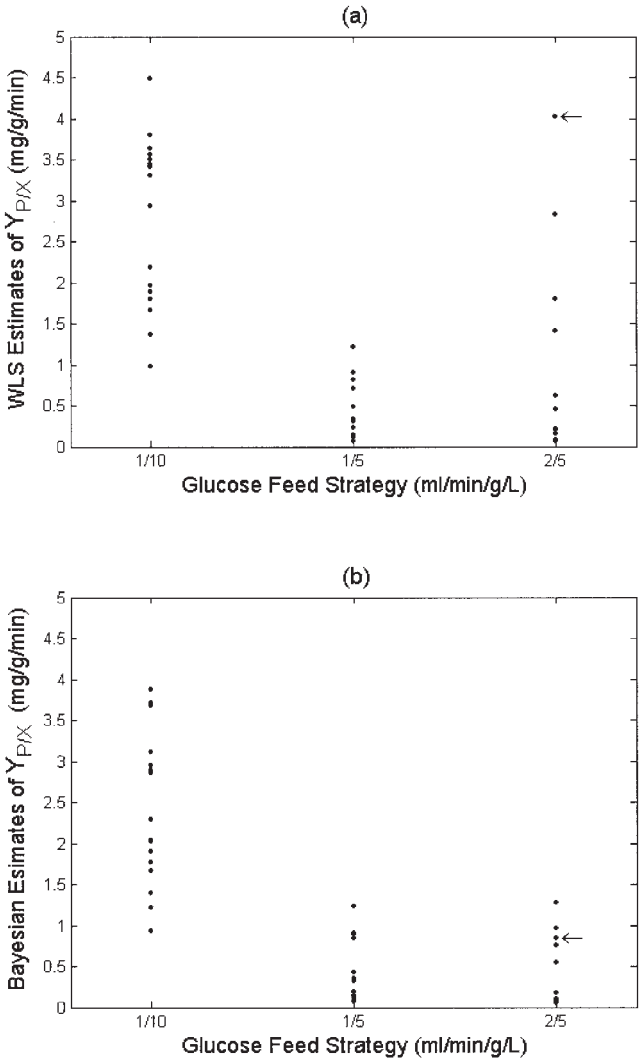

Figure 8. Estimated values of the stoichiometric coefficient $Y_{P/X}$ vs. the feed strategy.

(a) Corresponds to estimated values of $Y_{P/X}$ found via WLS, and (b) corresponds to estimated values of $Y_{X/P}$ found by taking the mean values of the Bayesian posterior. Each data point represents the estimated value of $Y_{X/[\text{infi}]P}$ of an experiment with a unique combination of fermentation inputs. The arrowed point in (a) was estimated from the same data set as the arrowed point in (b). It can be seen that the Bayesian estimate is much lower than the WLS estimate. Several other estimated values of $Y_{P/X}$ at high glucose feed strategies significantly changed when the Bayesian approach was taken.

with arrows represent the same fermentation. Data for this fermentation are shown in Figure 9. In this Figure, it can be seen that sufficient data were not collected to estimate $Y_{P/X}$ and t_{lag} with a great deal of certainty. Implementing informative priors in a Bayesian posterior defined preferences for a more gradual decrease in growth during the death phase, a slower consumption of inducer (which prolongs the amount of time that protein is being produced), and a shorter lag time between the production of protein and the observability of protein (which indirectly specifies a preference for smaller values of $Y_{P/X}$). We do not have sufficient data to prove that the dashed curve shown in Figure 9 is closer to the truth than the solid curve. However, we can show that the bias implemented upon all the fermentation model parameters improved our ability to predict future fermentation kinetics.

Discussion

In this article we present Bayesian methods of including prior information into the estimation of parameters of nonlinear dynamic systems. The implementation of prior knowledge here utilizes hyper-parameters to shrink parameter estimates towards more reasonable values. Tikhonov[16] developed non-Bayesian methods to shrink parameter estimates in situations where data are sparse and noisy; these methods use heuristic approaches (e.g., cross validation, trial and error) to determine optimal values of regularization parameters (a regularization parameter is similar to a hyper-parameter used in the context of Tikhonov methods). Nounou et al.[39] discuss Bayesian methods to determine optimal hyper-parameter values in the context of

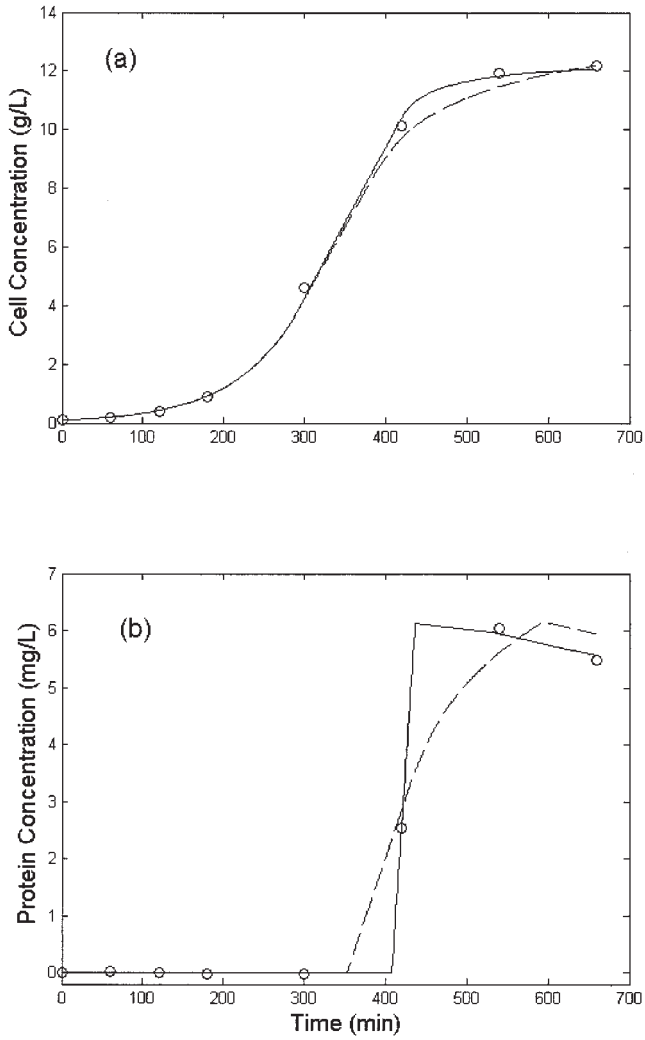

Figure 9. Data observations and model fits for data set corresponding to arrowed point of Figure 9.

Solid lines represent simulations based upon the WLS parameter estimates. Dashed lines represent simulations based upon mean values of the Bayesian posterior. The WLS estimates fit the data better. The Bayesian estimates predict more gradual production of protein.

linear process models; such methods are referred to as empirical Bayesian approaches.[26]

In this article a fully Bayesian approach is taken; a fully Bayesian approach requires that prior distributions be assigned to all hyper-parameters before any data have been observed. The posterior is then determined by integrating over all possible values of the hyper-parameter. The main advantage in this approach is that it can efficiently make use of a wide variety of data sets without having to alter the parameter estimation algorithm. If data are sparse or noisy, the hyper-parameter values will increase the influence of the prior information and improve parameter estimates (if the specified prior information is accurate). If enough quality data are available, the influence of the prior information should be insignificant. The main challenges in this approach are determining prior distributions for each parameter and effectively integrating the posterior. Forming good priors for model, noise, and hyper-parameters can be a difficult task, especially when the practitioner is not entirely sure how to quantify his beliefs. When no prior knowledge is available, non-informative priors may be used; however, such priors are also improper (illustrated in Figure 1) and result in undesirable results. For example, Figure 3d shows a situation where a non-informative prior over a hyper-parameter causes unreasonable shrinkage of the estimated model parameter. A similar phenomenon would occur in case study two if a non-informative prior were used for $\log(\sigma_{K_S})$. In most cases it should not be too difficult to determine weakly informative priors that consist of upper and lower bounds. Helpful prior information may be more laborious to quantify but may be worth the extra effort if the available data are limited. For example, in case study three, t_{lag} was a difficult parameter to estimate. DeLisa et al.[20] found t_{lag} to be approximately 95 (min) at 30°C, while Heim et al.[40] found t_{lag} to be approximately 240 (min) at 22°C; however, it is expected that changing other experimental conditions will also alter the value of t_{lag}. The operating range of temperature for our experimental *E. coli* database was 30-37°C; thus, we believe that t_{lag} is likely be closer to 95 (min) than 240 (min). This information helped to form the prior distributions over t_{lag} and $\sigma_{t_{lag}}$.

The overall theme to this article has been that minimizing the error between observed data and predictions of a model (e.g., weighted least squares) does not always lead to optimal results in parameter estimation; the inclusion of prior information alters what is perceived as a "best" fit solution and often such prior information should not be ignored, especially when data are sparse or noisy. The Bayes Theorem is an effective means of using simple laws of probability to combine prior information with information from observed data.

Acknowledgments

The authors would like to thank the American Vineyard Foundation for partial support of this work and Professor Wesley Johnson for useful discussions on the implementation of Bayesian methods.

Notation

Variables

C_I = concentration of inducer in feed
$\boldsymbol{D}$ = observed data
e_{ji} = random error added to y_{ji} to simulate $\tilde{y}_{ji}$ in case study two
F_I = feed rate of inducer (mL/min)
F_S = feed rate of substrate (mL/min)
$\bar{g}$ = mean value of observed product yields
g_i = observed ith product yield
i, j, or k = arbitrary index
I = inducer (arabinose) concentration (mg/L)
K_D = deceleration constant of growth rate in the death phase (L/g)
K_P = product inhibition constant (g/L)
K_S = monod constant (g/L)
N = number of data points
N_S = number of MCMC samples
N_P = number of parameters in posterior (model, noise, and hyper)
P = product or protein concentration (g/L or mg/L)
P_f = fluorescing protein (mg/L)
P_0 = initial product concentration
q_I = inducer consumption rate (mg/L/min)
R = correlation coefficient
$\hat{R}_j$ = potential scale reduction factor for MCMC estimate of j^{th} parameter
S = substrate concentration (g/L)
S_0 = initial substrate concentration (g/L)
t = time (min)
t_{lag} = time lag associated with produced and fluorescing protein (min)
u = simulated random variable uniformly distributed between zero and one
V = volume (L)
$WSSE$ = weighted sum of squared error criterion
X = cell concentration (g/L)
X_D = cell concentration where linear growth phase ends (g/L)
X_L = cell concentration where growth shifts from exponential to linear (g/L)
X_0 = initial cell concentration
$\tilde{\boldsymbol{Y}}$ = matrix of all observed state variables
y_{ji} = simulated j^{th} state variable at time t_i
$\tilde{y}_{ji}$ = observed j^{th} state variable at time t_i
$Y_{P/X}$ = stoichiometric coefficient (product from cell mass)
$Y_{X/S}$ = stoichiometric coefficient (cell mass from substrate)
α_j = signal to noise ratio for the j^{th} state variable
β = mean product yield (case study one)
$\beta_{\max}$ = maximum allowable value of β
$\beta_{\min}$ = minimum allowable value of β
$\boldsymbol{\varepsilon}$ = random vector used to perturb current MCMC sample values
$\boldsymbol{\phi}$ = vector of model parameters
μ = growth rate (h^{-1})
μ_{max} = maximum specific growth rate (h^{-1} or s^{-1})
$\boldsymbol{\theta}$ = vector of all model, noise, and hyper parameters
$\boldsymbol{\Theta}$ = $N_S \times N_P$ MCMC matrix representing a discrete approximation to $\Pr(\boldsymbol{\theta}|\boldsymbol{D})$
σ = standard deviation of error in observed data
σ_β = hyper-parameter corresponding to a half Gaussian prior distribution over β.
σ_{K_S} = hyper-parameter corresponding to a half Gaussian prior distribution over K_S.
Σ_q = covariance matrix of MCMC proposal distribution (used to generate values of $\boldsymbol{\varepsilon}$)
τ = protein production rate (h^{-1})

Functions and probability distributions

$E[f(\theta)]$ = expectation of arbitrary function
$f_j(\phi, t_i)$ = simulated j^{th} state variable at time t_i (solved by integrating ODE model with $\boldsymbol{\phi}$)
$L(\theta|\boldsymbol{D})$ = likelihood function
$N(\boldsymbol{\beta},\boldsymbol{\sigma})$ = normal distribution with mean of $\boldsymbol{\beta}$ and standard deviation of $\boldsymbol{\sigma}$
$Pr(\boldsymbol{D})$ = marginal likelihood
$Pr(\boldsymbol{D}|\boldsymbol{\theta})$ = sampling distribution
$Pr(\boldsymbol{\theta})$ = prior probability distribution
$Pr(\boldsymbol{\theta}|\boldsymbol{D})$ = posterior probability distribution

Acronyms

GFP = green fluorescent protein
$MCMC$ = Markov Chain Monte Carlo

NN = neural network
ODE = ordinary differential equation
WLS = weighted least squares

Literature Cited

1. vanBoekel MAJS. Statistical aspects of kinetic modeling for food science problems. *J Food Sci.* 1996;61:477-485.
2. Donaldson JR, Schnabel RB. Computational experience with confidence-regions and confidence-intervals for nonlinear least-squares. *Technometrics.* 1987;29:67-82.
3. Mendes P, Kell DB. Non-Linear optimization of biochemical pathways: applications to metabolic engineering and parameter estimation. *Bioinformatics.* 1998;14:869-883.
4. Bard Y. *Nonlinear Parameter Estimation*. New York: Academic Press; 1974.
5. Bates DM, Watts DG. *Nonlinear Regression Analysis and Its Applications*. New York: Wiley; 1988.
6. Box GEP, Draper NR. The Bayesian estimation of common parameters from several responses. *Biometrika.* 1965;52:355-365.
7. Gelman A, Carlin JB, Stern HS, Rubin DB. *Bayesian Data Analysis*. Boca Raton, Fla: Chapman & Hall/CRC; 2004.
8. Tierney L. Markov-Chains for exploring posterior distributions. *Ann Stat.* 1994;22:1701-1728.
9. Besag J, Green P, Higdon D, Mengersen K. Bayesian computation and stochastic-systems. *Stat Sci.* 1995;10:3-41.
10. Liu JS. *Monte Carlo Strategies in Scientific Computing*. New York: Springer; 2001.
11. Chen WS, Bakshi BR, Goel PK, Ungarala S. Bayesian estimation via sequential Monte Carlo sampling: unconstrained nonlinear dynamic systems. *Ind Eng Chem Res.* 2004;43:4012-4025.
12. Bois FY, Fahmy T, Block JC, Gatel D. Dynamic modeling of bacteria in a pilot drinking-water distribution system. *Water Res.* 1997;31: 3146-3156.
13. Pouillot R, Albert I, Cornu M, Denis JB. Estimation of uncertainty and variability in bacterial growth using bayesian inference. application to listeria monocytogenes. *Int J Food Microbiol.* 2003;81:87-104.
14. Shuler ML, Kargi F. *Bioprocess Engineering*. Upper Saddle River, NJ: Prentice Hall; 2002.
15. Blanch HW, Clark DS. *Biochemical Engineering*. New York: M. Dekker; 1996.
16. Tikhonov AN, Arsenin Vë. *Solutions of Ill-Posed Problems*. Washington: Halsted Press; 1977.
17. Mackay DJC. Bayesian interpolation. *Neural Comput.* 1992;4:415-447.
18. Wang JB, Zabaras N. A Bayesian inference approach to the inverse heat conduction problem. *Int J Heat Mass Tran.* 2004;47:3927-3941.
19. Albano CR, Randerseichhorn L, Bentley WE, Rao G. Green fluorescent protein as a real time quantitative reporter of heterologous protein production. *Biotechnol Progr.* 1998;14:351-354.
20. DeLisa MP, Li JC, Rao G, Weigand WA, Bentley WE. Monitoring Gfp-operon fusion protein expression during high cell density cultivation of Escherichia coli using an on-line optical sensor. *Biotechnol Bioeng.* 1999;65:54-64.
21. Buck KKS, Subramanian V, Block DE. Identification of critical batch operating parameters in fed-batch recombinant E-coli fermentations using decision tree analysis. *Biotechnol Progr.* 2002;18:1366-1376.
22. Coleman MC, Buck KKS, Block DE. An integrated approach to optimization of Escherichia coli fermentations using historical data. *Biotechnol Bioeng.* 2003;84:274-285.
23. Bretthorst GL. An introduction to parameter estimation using Bayesian probability theory. In: Fougère PF, ed. *Maximum Entropy and Bayesian Methods*. Dordrecht: Kluwer Academic Publishers; 1990:53-79.
24. Christensen R, Hanson T, Johnson W. *Bayesian Ideas and Data Analysis*. Unpublished manuscript, 2004.
25. Loredo TJ. From Laplace to supernova Sn 1987a: Bayesian inference in astrophysics. In: Fougère PF, ed. *Maximum Entropy and Bayesian Methods*. Dordrecht: Kluwer Academic Publishers; 1990:81-142.
26. Carlin BP, Louis TA. *Bayes and Empirical Bayes Methods for Data Analysis*. Boca Raton: Chapman & Hall/CRC; 2000.
27. Jeffreys H. *Theory of Probability*. Oxford: The Clarendon Press; 1939.
28. Box GEP, Tiao GC. *Bayesian Inference in Statistical Analysis*. Reading, Mass: Addison-Wesley Pub. Co.; 1973.
29. MacKay DJC. *Information Theory, Inference, and Learning Algorithms*. Cambridge, UK and New York: Cambridge University Press; 2003.
30. Press WH, Teukolsky SA, Vettering WT, Flannery BP. *Numerical Recipes in C*. Cambridge, England: Cambridge University Press; 1996.
31. Metropolis N, Rosenbluth AW, Rosenbluth MN, Teller AH, Teller E. Equations of state calculations by fast computing machines. *Journal of Chemical Physics.* 1953;21:1087-1092.
32. Chib S, Greenberg E. Understanding the Metropolis-Hastings algorithm. *Am Stat.* 1995;49:327-335.
33. Roberts GO, Gelman A, Gilks WR. Weak convergence and optimal scaling of random walk Metropolis algorithms. *Ann Appl Probab.* 1997;7:110-120.
34. Cowles MK, Carlin BP. Markov Chain Monte Carlo convergence diagnostics: a comparative review. *J Am Stat Assoc.* 1996;91:883-904.
35. Kass RE, Carlin BP, Gelman A, Neal RM. Markov Chain Monte Carlo in practice: a roundtable discussion. *Am Stat.* 1998;52:93-100.
36. Thompson ML, Kramer MA. Modeling chemical processes using prior knowledge and neural networks. *Aiche J.* 1994;40:1328-1340.
37. Nabney I. *Netlab: Algorithms for Pattern Recognition*. London and New York: Springer; 2002.
38. Reich Y, Barai SV. Evaluating machine learning models for engineering problems. *Artif Intell Eng.* 1999;13:257-272.
39. Nounou MN, Bakshi BR, Goel PK, Shen XT. Process modeling by Bayesian latent variable regression. *Aiche J.* 2002;48:1775-1793.
40. Heim R, Prasher DC, Tsien RY. Wavelength mutations and posttranslational autoxidation of green fluorescent protein. *P Natl Acad Sci USA.* 1994;91:12501-12504.

Manuscript received Oct. 26, 2004, and revision received July 12, 2005.