“Cool Runnings” means “Peace Be The Journey.”

Derice Bannock, Cool Runnings

University of Alberta

Non-Linear Reparameterization of Complex Models
with Applications to a Microalgal Heterotrophic Fed-Batch Bioreactor

by

Kartik Surisetty

A thesis submitted to the Faculty of Graduate Studies and Research
in partial fulfillment of the requirements for the degree of

Master of Science
in
Process Control

Department of Chemical and Materials Engineering

Spring 2010
Edmonton, Alberta

Examining Committee

Amos Ben-Zvi, Chemical and Materials Engineering

Randall Weselake, Agricultural, Food and Nutritional Science

Vinay Prasad, Chemical and Materials Engineering

This thesis is dedicated to the people who have been with me every step of the way, through good times and bad. MOM and DAD, thank you for your endless love, guidance, and unconditional support that you have given me over the years, giving me confidence and helping me succeed in life. I am what I am because of you… I Love You.

To my wonderful parents:
Venkataraman and Padmavathi Surisetty

Thank you for everything

Your son:
Kartik Surisetty

Abstract

Good process control is often critical for the economic viability of large-scale production of several commercial products. In this work, the production of biodiesel from microalgae is investigated. Successful implementation of a model-based control strategy requires the identification of a model that properly captures the biochemical dynamics of microalgae, yet is simple enough to allow its implementation for controller design. For this purpose, two model reparameterization algorithms are proposed that partition the parameter space into estimable and inestimable subspaces. Both algorithms are applied using a first principles ODE model of a microalgal bioreactor, containing 6 states and 12 unknown parameters. Based on initial simulations, the non-linear algorithm achieved better degree of output prediction when compared to the linear one at a greatly decreased computational cost. Using the parameter estimates obtained through implementation of the non-linear algorithm on experimental data from a fed-batch bioreactor, the possible improvement in volumetric productivity was recognized.

Acknowledgements

I would like to thank all the people who have helped and inspired me during my Masters study. It has been an honour to work with you and my Masters experience would have been incomplete without you.

The guidance provided by my advisor, Dr. Amos Ben-Zvi, during my research and study at the University of Alberta is invaluable. His energy and enthusiasm motivated me to do my best. He is always available and ready to help his students; as a result, my research time went by smoothly and was one of the most rewarding parts of my life. I would like to give my heartfelt gratitude for his time and effort. It has been my utmost honour working with him.

Sincere thanks needs to be expressed to Hector De la Hoz Siegler without whom I could not have successfully completed my project. Working with him was fun, enriching, and entertaining. He is smart, honest and a delightful friend.

Special thanks to all my friends Qi, Hector, Vivek, Hua, Yan, Xing, Yu, Venkat, and Imran for all the fun we've had during our time together. The trips to Banff, Vancouver, Montreal, Elk Island and Jasper will be one of the most memorable times of my Masters experience. There are no words to express my immense gratefulness for you letting me be your friend. I would like to especially thank Imran and Hector for reviewing my thesis. I look forward to all the fun times in store for our friendship.

Finally, I cannot end without conveying my deepest gratitude for my parents, brother, grandparents, cousins, aunts and uncles for their endless love and encouragement. You are the light of my life.

Contents

List of Figures

List of Tables

1

Introduction

Modeling is the practice of approximating physical reality using an acceptable mathematical form. A model is a mathematical representation that, in essence, gives a relation between system inputs and outputs. There are usually two types of models: empirical and physical models. Empirical models are based on the black-box concept while the physical models are based on an understanding of the internal system dynamics and the interactions between the system and the surroundings (Elnashaie and Garhyan 2003). Models obtained from mass, energy and momentum balance equations are good examples of physical models. In this work a first-principles based microalgal bioreactor model is used.

Modeling a chemical or biochemical system significantly depends on the *a priori* knowledge of the physical and chemical laws that govern the processes that take place within the boundaries of the system (Elnashaie and Garhyan 2003). Mass, heat and momentum transfer rates, reaction rates, adsorption-desorption rates, thermodynamic limitations are some of the variables that one has to take into account while modeling a chemical or biochemical system. Consequently, physicochemical parameters need to be identified in order to obtain a model output for a certain input. In reality, one may not be certain of these parameter values; therefore, before using the model for control and optimization, one must validate the model against the real system.

In this work, the production of biodiesel using microalgae is investigated. Worldwide interest in biofuels has been increasing in recent times due to issues such as climate change and security of energy supply (Peters and Thielmann 2008). The US Environmental Protection Agency (USEPA) found that there is 67% reduction of green house gases emissions when biodiesel is used as an alternative to traditional fossil fuels

(USEPA 2002). Moreover, the use of biodiesel decreases tailpipe particulate matter (-10.1%), unburned hydrocarbon (-21.1%) and carbon monoxide (-11%) emissions when compared to emissions from regular diesel fuel (USEPA 2002). Biodiesel being biodegradable and non-toxic further increases its suitability of being used as an alternative fuel. A major obstacle for biodiesel to replace current energy demand is limited acreage availability (Christi 2007). Producing biodiesel from crops such as corn and soybeans uses land that has traditionally been used for food production (Peters and Thielmann 2008).

Microalgae are a feasible alternative to crops for the production of biodiesel. They are highly efficient biological organisms with higher biomass production and faster growth rate compared to other energy crops ((Minowa, et al. 1995), (Miao and Wu 2006)). Microalgae produce and store high amounts of oil, which can be used as either a precursor to biodiesel (Li, Xu and Wu 2007) or as single cell oils (Chen and Chen 2006). Oil content in microalgae ranges from 11% to 77% (Christi 2007) with variations occurring due to strain genetics and culture conditions. Moreover, microalgae can be produced in desertic areas or intensively in bioreactors and therefore do not require the use of land that would otherwise be used for food crops.

Currently, biodiesel production using microalgae is not economically competitive compared to production from conventional sources. Increasing the algal growth rate and oil accumulation rate will significantly improve the economic competitiveness of algal-based biodiesel production. Optimal model-based control strategy is a promising option in controlling culture conditions in order to achieve good algal growth and oil production rates. A reliable model of a microalgal bioreactor system needs to be obtained to successfully implement a model-based control strategy, as microalgal systems are highly non-linear in nature. Parameters of microalgal models cannot be directly obtained from measurement (Audoly, et al. 2001); therefore, estimating these parameters requires model reparameterization and well designed experiments.

A major limitation in obtaining a reliable model is parameter inestimability. If the data collected for parameter identification and model validation is not adequate to accurately estimate every parameter, then the corresponding mathematical model is considered to be inestimable. Inestimability implies that several parameter values will lead to statistically

indistinguishable predictions (Ben-Zvi 2008). Significant amount of literature is present on identifying and dealing with inestimable parameters ((Yao, et al. 2003); (Ben-Zvi, McAuley and McLellan 2004); (Sidoli, Mantalaris and Asprey 2005)). There are two options available if a model is inestimable: obtain additional data and fix inestimable parameters at some nominal value. Investing in obtaining additional data is not always cost effective as the degree of improvement in model predictions might not be enough to justify the additional cost. Therefore, one is often interested in estimating only a subset of the unknown model parameters.

Parameter estimability of a first-principles based microalgal bioreactor model consisting of six ordinary differential equations and 12 unknown parameters is studied in this work. It is seen that, even when using an optimal experimental design, many of the model parameters are inestimable. Two model reparameterizing algorithms are proposed in order to reduce the number of parameters that must be estimated for accurate predictions. Both the algorithms partition the parameter space into estimable and inestimable subspaces. For the linear reparameterization, singular value decomposition of the parameter covariance matrix is used to identify a set of four directions in the twelve dimensional parameter space along which significant change in the output occurs. In the nonlinear reparameterization algorithm, the three system rate functions are utilized as pseudo-outputs in order to perform a nonlinear transformation which reduces the dimension of the parameter space from twelve to three. Based on initial simulations, both algorithms greatly decrease computational time while achieving a good degree of output prediction and significantly reduce computational complexity. Therefore, due to its lower Weighted Sum of Squared Error, WSSE, and lower computational cost, the non-linear algorithm is better than its linear counterpart.

In order to verify the potential of the proposed non-linear model reparameterization algorithm, it was implemented using actual bioreactor data. The experiment was performed using input profiles designed using the D-optimality criterion with respect to the three pseudo-parameters in the non-linear algorithm. Total algal oil stored in cells, external nitrogen source concentration, external carbon source concentration, and total biomass concentration were measured as experimental data. The algorithm had to be modified by introducing an iterative step, in order to achieve a good degree of output prediction. This iterative step allowed for updating the non-linear transformation of the

parameter space, as improved parameter estimates became available. The iterative non-linear algorithm is able to greatly decrease computational time while achieving a high degree of prediction accuracy.

1.1 Objectives and contributions

The main goal of this work is to develop an algorithm that can be used to reparameterize non-linear differential equation models while maintaining their prediction capability. The proposed algorithm is applied to a microalgal bioreactor system. The proposed algorithm should achieve a good degree of output prediction while considerably decreasing computational cost and time when compared to the case where all model parameters are estimated. The following are the stage-wise objectives of the work:

1. Design an input profile that is optimal with respect to the 12 model parameters.
2. Employ a perturbation-based test to investigate which parameters have negligible effect on model predictions.
3. Develop a model reparameterization algorithm that is based on identifying directions in the parameter space along which there is significant change in model predictions.
4. Use the transformation-based approach for reparameterizing inestimable systems proposed by Ben-Zvi (2008) to develop a non-linear model reparameterization algorithm.
5. Utilize simulations, based on the designed optimal input profiles, to compare the performance of the two proposed model reparameterization algorithms.
6. Develop an experimental design scheme to obtain an input profile that is optimal with respect to the three pseudo-parameters in the non-linear algorithm.
7. Perform an experiment by implementing the designed input profile (objective 6) on an actual fed-batch microalgal bioreactor.
8. Collect and analyze samples in order to obtain measured values.
9. Apply non-linear model reparameterization algorithm using the measured bioreactor data.

The following are the significant contributions of the work:

1. Collaborated with co-authors to develop a novel model for a microalgal bioreactor system.
2. Performed an estimability study of the model, leading to the conclusion that there exist directions in the parameter space along which minimal change in the output occurs.
3. Developed a method to obtain a linear transformation that partitions the parameter space into estimable and inestimable subspaces.
4. Extended the approach proposed by Ben-Zvi (2008) to cases where one cannot analytically integrate the pseudo-parameters. This was done by using Taylor series approximation and numerical integration. This led to the non-linear model reparameterization algorithm.
5. Compared the linear and non-linear model reparameterization algorithms through simulations. The non-linear algorithm achieved a higher degree of output prediction while greatly decreasing computational cost.
6. Improved on the non-linear algorithm by introducing an iterative step where the non-linear transformation is updated as better parameter estimates become available.
7. Validated the model with experimental data and provided parameter estimates.
8. Developed an approach for the optimal experimental design with the objective of obtaining optimal estimates for only the estimable parameter combinations.

1.2 Thesis outline

In this thesis, a model reparameterization algorithm is proposed for the purpose of output prediction for a microalgal fed-batch bioreactor. **Chapter 2** contains the background information on the topics that are closely related to this work. Topics such as models used for a microalgal bioreactor and motivation for model reparameterization. **Chapter 3,** Paper 1**,** contains the derivation and comparison between the two proposed model reparameterization algorithms. **Chapter 4**, Paper 2, shows the results of applying the non-linear model reparameterization algorithm on measured data from an actual microalgal fed-batch bioreactor. **Chapter 5** summarizes all the results of this work and

presents directions for future work. **Appendix A** contains all the MATLAB® and MAPLE™ code pertaining to the non-linear model reparameterization algorithm. **Appendix B** contains the experimental data. Note that there is overlap between several chapters as this thesis is compiled according to the paper-format guidelines given by the Faculty of Graduate Studies and Research (FGSR) at the University of Alberta.

References

Audoly, Stefania, Guiseppina Bellu, Leontina D'Angiò, Maria Pia Saccomani, and Claudio Cobelli. "Global identifiability of nonlinear models of biological systems." *IEEE Transactions on Biomedical Engineering* 48, no. 1 (January 2001).

Ben-Zvi, Amos. "Reparameterization of inestimable systems with applications to chemical and biochemical reactor systems." *AIChE Journal* 54, no. 5 (2008): 1270-1281.

Ben-Zvi, Amos. "Reparameterization of inestimable systems with applications to chemical and biochemical reactor systems." *AIChE Journal* 54, no. 5 (2008): 1270-1281.

Ben-Zvi, Amos, Kim McAuley, and James McLellan. "Identifiability study of a liquid–liquid phase-transfer catalyzed reaction system." *AIChE Journal* (Wiley InterScience) 50 (2004): 2493–250.

Chen, G.Q., and F Chen. "Growing phototrophic cells without light." *Biotechnology Letters* 28 (2006): 607-616.

Christi, Y. "Biodiesel from microalga." *Biotechnology Advances* 25 (2007): 294-306.

Elnashaie, Said S.E.H., and Parag Garhyan. *Conservation equations and modeling of chemical and biochemical processes.* Aurburn, Alabama: Marcel Dekker, Inc., 2003.

Li, X, H Xu, and Q Wu. "Large-scale biodiesel production from from microalga *Chlorella protothecoides* through heterotrophic cultivation in bioreactors." *Biotechnology and Bioengineering* 4 (2007): 764-771.

Miao, X, and Q Wu. "Biodiesel production from heterotrophic microalgal oil." *Bioresource Technology* 97 (6) (2006): 841-846.

Minowa, T, S Yokoyama, M Kishimoto, and T Okakura. "Oil production from algal cells of *Dunaliella tertiolecta* by direct thermochemical liquefaction." *Fuel* 74, no. 12 (1995): 1735-1738.

Peters, Jörg, and Sascha Thielmann. "Promoting biofuels: implications for developing countries." *Energy Policy* 36 (2008): 1538–1544.

Sidoli, FR, A Mantalaris, and SP Asprey. "Toward global parametric estimability of a large-scale kinetic single-cell model for mammalian cell cultures." *Industrial & Engineering Chemistry Research* 44, no. 4 (Febuary 2005): 868-878.

USEPA. "A comprehensive analysis of biodiesel impacts on exhaust emissions." United States Environmental Protection Agency., 2002, EPA 420-P-02-001.

Yao, ZK, BM Shaw, KB McAuley, and DW Bacon. "Modeling ethelyne/butene copolymerization with multi-site catalysts: Parameter estimability and experimental design." *Polymer Reaction Engineering* 11 (2003): 563-588.

2

Background

2.1 Chapter overview

This chapter contains the relevant background information pertaining to this thesis. **Section 2.2** discusses the production of biodiesel from microalgae. **Section 2.3** introduces different models and kinetic rate functions used to model bioreactors. In **Section 2.4**, the concepts of identifiability and estimability are presented and the motivation for model reparameterization is discussed. In **Section 2.5**, other topics, such as singular value decomposition, that are utilized in this work are presented.

2.2 Modeling of a bioreactor

A key variable in evaluating the dynamics of a microalgal bioreactor is biomass concentration (X). In order to control biomass concentration, one can manipulate the nutrient concentration in the media (*S*). Over the years, many models, that relate nutrient concentration to biomass concentration, have been proposed.

2.2.1 Classic growth models

The Monod model is one of the classic models used to model growth rate in a bioreactor. Monod model is an empirical equation that simulates the cell growth rate in terms of the external concentration of the limiting substrate. The equation is a generalization of the Michaelis-Menten kinetic expression for enzymatic systems. If there is a causal relationship between nutrient exhaustion and end of growth then the nutrient is said to be

limiting (Lobry, et al. 1992). An important characteristic of the Monod behaviour is that there is an upper limit to growth rate when the nutrient is in great excess and there is no growth when the nutrient concentration is zero. Using the Monod model, the biomass growth rate, μ, is given as follows (Monod 1949):

$$\mu = \mu_m \left(\frac{S}{K_s + S} \right) \tag{2.1}$$

where μ_m is the maximum growth rate and K_s is the nutrient concentration that supports half the maximum growth rate. Note that using Monod model; the growth rate is dependent on nutrient concentration in the media and not the concentration inside the cell.

The Haldane model is similar to the Monod model but with the addition of nutrient inhibition. Therefore, unlike Monod behaviour where there is a maximum growth rate at excess nutrient concentration, the growth rate decreases after a certain nutrient concentration. This implies that there is an optimal nutrient concentration at which the maximum growth occurs. Based on the Haldane model, it can be concluded that running the bioreactor in excess nutrient does not achieve the best performance with respect to biomass growth. Using the Haldane kinetics the biomass growth rate, μ, is given as follows (Wang, Krstic and Bastin 1999):

$$\mu = \mu_m \left(\frac{S}{S + K_s + \frac{S^2}{K_I}} \right) \tag{2.2}$$

where K_I is the inhibition constant. Even in the Haldane kinetics, the growth rate is dependent on extracellular nutrient concentration.

2.2.2 Cell-quota models

The Monod and Haldane models have been widely used to model bacterial bioreactors; however, there is a clear difference in the dynamics of bacterial and algal systems. Microalgae exhibit a phenomenon called “luxury consumption” that is the initial uptake

rates of a nutrient are far in excess of the organism's growth rate (Droop 1973). This is evident for nutrients such as phosphorus and nitrogen. In order to model this behaviour, the intracellular nutrient quota (Q) is introduced as an intermediate state variable, in order to distinguish between nutrient uptake rate, ρ, and nutrient-controlled growth rate.

Droop model is the first proposed quota model and it is different from Monod model because it takes into account the notion of an internal nutrient pool. The growth and nutrient uptake rates using the Droop model are given below (Droop 1973):

$$\mu = \mu'_m \left(1 - \frac{k_Q}{Q}\right) \tag{2.3}$$

$$\rho = \rho_m \left(\frac{S}{k_s + S}\right) \tag{2.4}$$

where μ'_m differs from μ_m in Equation 2.1, as it is the maximum growth rate based on intracellular nutrient quota (Q) and not the nutrient concentration in the media (S). ρ_m is the maximum uptake rate, k_s is the nutrient concentration that supports half the maximum uptake rate, and k_Q is the subsistence quota.

Another quota model was proposed by Caperon (1972). The Caperon model introduces concept of minimum quota required for growth and minimum extracellular nutrient concentration for nutrient uptake. The growth and nutrient uptake rates using the Caperon model are given below (Caperon and Meyer 1972):

$$\mu = \mu'_m \left(\frac{Q - Q_0}{K_q + (Q - Q_0)}\right) \tag{2.5}$$

$$\rho = \rho_m \left(\frac{S - S_0}{K'_S + (S - S_0)}\right) \tag{2.6}$$

where $\mathrm{Q_0}$ is the minimum nutrient quota at zero growth rate and $\mathrm{K_q}$ is the half saturation constant of nutrient quota for growth. S_0 is the nutrient concentration at which uptake rate is zero and K'_S is the half saturation constant of extracellular nutrient concentration for

nutrient uptake. The behaviour of cell-quota models is investigated in detail by Tett (1988).

2.2.3 Model selection

Algal growth models are often classified based on the number of compartments they contain, i.e. the number of different nutrients the model considers. There is a cost associated with considering more compartments as each rate determining step for each nutrient must contain at least two parameters and all of these parameters need to be estimated for the model to have any practical applications (Tett and Droop 1988). The number of nutrients is usually decided based on a compromise between model accuracy and robustness. Typically, a model is considered sufficiently accurate if it is satisfactorily able to predict a particular set of outputs trajectories and robust if it is able to predict a different set of output without significant loss of accuracy. The concepts of model accuracy and robustness naturally conflict and a compromise must be based on what are the objectives of the work or the purpose of the model (Tett and Droop 1988).

2.3 Motivation of model reparameterization

In this work, a first principles microalgal bioreactor model containing six states, three inputs, four outputs, and twelve parameters is considered. Since, there are a significant number of model parameters in this highly non-linear model, parameter identifiability and estimability must be considered before proceeding with parameter estimation.

2.3.1 Identifiability

Good estimates of model parameters are required before the model can be used for practical purposes, such as design, scale-up, control, and optimization. If the same input-output set can be explained by different sets of parameter values, applying the model for design and control may be challenging. Therefore, model identifiability must be checked *a priori* as the cause for model unidentifiability is a flaw in the model structure formulation (Ben-Zvi 2008) and even a large number of experiments will not lead to unique parameter estimates.

A model is identifiable if and only if there is a unique input-output set for each parameter set. Several tests for identifiability of non-linear models have been presented in the literature. The generating series approach (Walter and Pronzato 1996) and the linearization approach (Ben-Zvi, McLellan and McAuley 2006) are a few examples. In this work, the linearization approach was used and the proposed bioreactor model was determined to be locally identifiable. This implies that unique parameter estimates can be obtained under idealized conditions such as no noise in the system, no plant-model mismatch, and desired inputs can be perfectly actuated for performing as many experiments as are required (Ben-Zvi, McAuley and McLellan 2004). These are conditions that rarely occur in real-life; therefore, after performing the identifiability test, it is beneficial to investigate parameter estimability. Estimability is more related to whether each parameter can be accurately estimated from a data set obtained from real-life noisy experimental conditions (Yao, et al. 2003).

2.3.2 Inestimability

Model estimability is the measure of whether parameters can be computed accurately from a given data set and experimental conditions (Yao, et al. 2003). Model identifiability is a necessary condition for model estimability; however, it is not sufficient. Identifiability of model parameters means certain experimental sets exist from which the parameters can be estimated uniquely. On the other hand, some other experimental sets might result in data from which estimating unique parameter values may be challenging or even impossible. Another perspective on model estimability is whether a certain input profile has enough excitation to result in a rich enough date set to uniquely estimate parameters.

If model parameters are inestimable, then different parameter values lead to statistically identical predictions (Ben-Zvi 2008). The problem of inestimability has been well documented in the literature. For example, model inestimability is encountered in the fields of copolymerization with multi-site catalysts (Yao, et al. 2003), ecological systems (Marsili-Libelli, Guerrizio and Checchi 2003), and membrane fuel cells (Corrêa, et al. 2005). Parameter estimates from inestimable systems cannot be used for design, scale-up, control and optimization as they are usually inaccurate. In practice, model inestimability

is usually dealt with by obtaining additional data or modifying the model. Investing in obtaining additional data may not be practical if the improvement in model predictions might not be enough to justify the additional cost of doing more experiments. Therefore, it is more feasible to modify the model by reducing the number of parameters. Model reparameterization is an efficient approach for model reduction.

2.3.3 Model reparameterization

The main objective for model modification, to deal with inestimability, is to simplify the model to the point where each parameter (or parameter combination) has a noticeable effect on model predictions. Model modification of inestimable systems is an active area of research and the following are a few approaches proposed in literature:

1. Parameters that are determined *a priori* to have negligible effect on model predictions can be fixed at a certain nominal value (Chakravarti, Ray and Zhang 2001).
2. Parameter values can be fixed at their nominal values sequentially based on correlation information (Matos, Mattos Neto and Pinto 2001).
3. Use a sensitivity-based approach to identify parameters resulting in negligible or no effect on model behaviour and lump, discard or fix them at their nominal value (Yao, et al. 2003). This method can also be used to construct confidence regions for parameters (Marsili-Libelli, Guerrizio and Checchi 2003).
4. Estimate different parameter subsets at different times using a Bayes theorem-based scheme (Preisig 2007). A similar scheme based on Monte-Carlo-based approach has also been investigated (Corrêa, et al. 2005).
5. Manually adjust parameter values until an acceptable fit is obtained (Khare, Seavey and Liu 2002).

The above mentioned approaches are based on identifying a subset of parameters that are inestimable. However, the work done by Ben-Zvi (2008) shows that the parameter combinations are inestimable rather than a subset of parameters. In addition, the inestimability of these parameter combinations is strongly related to experimental conditions. Therefore, dealing with parameter inestimability by reparameterizing the

model using a transformation that partitions the parameter space into estimable and inestimable subspaces is a promising approach.

Partitioning non-linear systems using transformation has been well documented in the literature (Isidori 1989). Non-linear transformations have traditionally been used to partition non-linear systems into controllable/uncontrollable or observable/unobservable subsystems. However, they have been rarely used to partition the system into estimable/inestimable subsystems. An approach that applies differential geometry to develop a non-linear transformation that partitions the parameter space into estimable and inestimable parts is proposed by Ben-Zvi (2008). This method is used in this work to develop the non-linear model reparameterization algorithm.

2.4 Other topics

2.4.1 Singular value decomposition

Singular value decomposition (SVD) is a widely used technique for matrix decomposition of rectangular real or complex matrices. It is used for computing the pseudo-inverse, matrix approximation, determining rank and null space of a matrix. Let A be a $m \times n$ matrix, and the singular value decomposition of A is given below:

$$A = USV^T \tag{2.6}$$

where U is a $m \times m$ unitary matrix, V^T is the conjugate transpose of a $n \times n$ unitary vector, V, and S is a $m \times n$ diagonal matrix with nonnegative real numbers on the diagonal. The columns of V are the eigenvectors of $A^T A$ and form a set of orthonormal basis vector directions for A. Similarly, the columns of U are the eigenvectors of AA^T (Chan 1982). In this work, SVD was performed using MATLAB.

2.4.2 Experimental design

It is difficult to determine *a priori* which parameters have the greatest effect on the model predictions and hence can be estimated. Therefore, to maximize the number of estimable

parameters, one can implement an optimal experimental design. The general objective of optimal experimental design is to obtain an input profile, which results in data with the maximum amount of information for subsequent parameter estimation.

For a model containing several parameters, the mean of the parameter estimations is a vector and their variance is a matrix. The reciprocal of the variance-matrix is called the information matrix. Generally, minimizing variance means maximizing information; however, since the variance of the estimated parameters is a matrix, minimizing the variance is not straightforward. Several optimality criterions focus on minimizing the variance of the information matrix; however, they differ in the objective function that is minimized. For example, the A-optimality criterion minimizes the trace of the information matrix (Horesh, Haber and Tenorio 2008) implying the average variance is minimized. The E-optimality criterion is based on minimizing the lowest eigenvalue of the information matrix (Horesh, Haber and Tenorio 2008); this corresponds to a minimax approach. The D-optimality criterion minimizes the determinant of the information matrix, hence reducing the volume of the uncertainty ellipsoid of parameter estimates (Horesh, Haber and Tenorio 2008). In this work, the D-optimality criterion was chosen in order to design an input profile as it puts emphasis on the quality of parameter estimates.

References

Ben-Zvi, Amos. "Reparameterization of inestimable systems with applications to chemical and biochemical reactor systems." *AIChE Journal* 54, no. 5 (2008): 1270-1281.

Ben-Zvi, Amos, James P McLellan, and K B McAuley. "Identifiability of non-linear differential algebraic systems via a linearization approach." *The Canadian Journal of Chemical Engineering* 84 (2006): 590-596.

Ben-Zvi, Amos, Kim McAuley, and James McLellan. "Identifiability study of a liquid–liquid phase-transfer catalyzed reaction system." *AIChE Journal* (Wiley InterScience) 50 (2004): 2493–250.

Caperon, J, and J Meyer. "Nitrogen-limited growth of marine phytoplankton– II. Uptake kinetics and their role in nutrient limited growth of phytoplankton." *Deep-Sea Research* 19, no. 9 (1972): 619–632.

Chakravarti, S, W Harmon Ray, and Simon X Zhang. "Kinetic study of olefin polymerization with a supported metallocene catalyst. IV. Comparison of bridged and unbridged catalyst in gas phase." *Journal of Applied Polymer Science* 81 (2001): 1451-1459.

Chan, Tony F. "An improved algorithm for computing the singular value decomposition." *ACM Transactions on Mathematical Software (TOMS)* (ACM) 8, no. 1 (March 1982): 72-83.

Corrêa, , Jeferson M, Felix A Farret, Vladimir A Popov, and Marcelo G Simões. "Sensitivity analysis of the modeling parameters used in simulation of proton exchange membrane fuel cells." *IEEE Transactions on Energy Conversion* 20 (2005): 211-218.

Droop, M. R. "Some thoughts on nutrient limitation in algae." *Journal of Phycology* 9 (1973): 264-272.

Horesh, Lior, Eldad Haber, and Luis Tenorio. *Optimal experimental design for the large-scale nonlinear ill-posed problem of impedance imaging.* Technical report , Mathematics and Computer Science, Atlanta: EMORY UNIVERSITY, 2008.

Horesh, Lior, Eldad Haber, and Luis Tenorio. *Optimal experimental design for the large-scale nonlinear ill-posed problem of impedance imaging.* Technical report, Mathematics and Computer Science, Atlanta: Emory university, 2008.

Isidori, A. *Nonlinear Control Systems.* New York: Springer-Verlag, 1989.

Khare, NP, KC Seavey, and YA Liu. "Steady-state and dynamic modeling of commercial slurry high-density polyethylene (HDPE) processes." *Industrial & Engineering Chemistry Research* 41 (2002): 5601-5618.

Lobry, J. R., J. P. Flandrois, G Carret, and A Pave. "Monod's bacterial growth model revisited." *Bulletin of Mathematical Biology* 54, no. 1 (1992): 117-122.

Marsili-Libelli, S, S Guerrizio, and N Checchi. "Confidence regions of estimated parameters for ecological systems." *Ecological Modelling* 165 (2003): 127-146.

Matos, V, A G Mattos Neto, and J C Pinto. "Method for quantitative evaluation of kinetic constants in olefin polymerizations. I. Kinetic study of a conventional Ziegler–Natta catalyst used for propylene polymerizations." *Journal of Applied Polymer Science* 79 (2001): 2076-2108.

Monod, J. "The growth of bacterial cultures." *Annual Review of Microbiology* 3 (1949): 371-394.

Preisig, HA. "Incremental nonlinear parameter estimation in dynamical systems." *International the IFAC Symposium on Dynamics and Control of Process Systems.* 2007. 297-302.

Tett, P, and M Droop. *Cell quota models and planktonic primary production.* Vol. 2, in *Handbook of Laboratory Model Systems for Microbial Ecosystems*, by Julian W T Wimpenny, edited by J.W.T. Wimpenny, 177–223. CRC Press, 1988.

Walter, E, and L Pronzato. "On the identifiability and distinguishability of nonlinear parametric models." *Mathematics and Computers in Simulation* 42 (1996): 125-134.

Wang, Hsin-Hsiung, Miroslav Krstic, and Georges Bastin. "Optimizing bireactors by extremum seeking." *International Journal of Adaptive Control and Signal Processing* 13 (1999): 651-669.

Yao, ZK, BM Shaw, KB McAuley, and DW Bacon. "Modeling ethelyne/butene copolymerization with multi-site catalysts: Parameter estimability and experimental design." *Polymer Reaction Engineering* 11 (2003): 563-588.

3

Model Reparameterization and Output Prediction for a Bioreactor system[1]

Abstract

Microalgal bioprocesses are of increasing interest due to the possibility of producing fine chemicals, pharmaceuticals and biofuels. In this work, the parameter estimability of a first principles ODE model of a microalgal bioreactor, containing 6 states and 12 unknown parameters, is investigated. For this purpose, the system input trajectories are computed using the D-optimality criterion. Even using a D-optimal input, not all parameters were found to have a significant effect on model predictions. Linear and nonlinear transformations are used to partition the parameter space into estimable and inestimable subspaces. For the linear reparameterization, a set of four directions in the twelve dimensional parameter space, along which a significant change in the output occurs, are identified using singular value decomposition of the parameter covariance matrix. The nonlinear reparameterization utilizes the three system rate functions as pseudo-outputs in order to perform a nonlinear transformation which reduces the dimension of the parameter space from twelve to three. Both proposed reparameterization methods achieve a good degree of output prediction at a greatly decreased computational cost.

[1] A version of this chapter has been submitted to the Chemical Engineering Science journal.

3.1 Introduction

Global warming and depletion of fossil fuels has increased the need for cleaner and sustainable energy production. Biodiesel, a proven alternative fuel, provides a 67% reduction of green house gases emissions when compared to fossil fuels (USEPA 2002). Microalgae have the ability to produce large amounts of oil that can be directly used as high value single-cell oils (Chen and Chen 2006), or be converted into biodiesel (Li, Xu and Wu 2007).

Good control of culture conditions is critical for the economic viability of large-scale production of microalgae. Model-based control strategies have been successfully applied to biochemical reaction systems to improve their economic performance (Bastin and Dochain 1990). In order to implement a model-based control strategy, one must identify a model that properly captures the biochemical dynamics of microalgae, yet is simple enough to allow its implementation for controller design. Biochemical systems are highly non-linear, and models of such systems typically contain parameters that are not directly accessible to measurement (Audoly, et al. 2001). Therefore, the use of a model to develop a controller for microalgal processes requires the estimation of the model parameters using well designed experiments.

A mathematical model is not estimable if the data collected for parameter identification and model validation is not sufficient for the accurate estimation of every parameter in the model. Inestimability implies that several parameter values will lead to statistically indistinguishable predictions (Ben-Zvi 2008). A large amount of literature has been devoted to identifying and dealing with inestimable parameters ((Yao, et al. 2003); (Ben-Zvi, McAuley and McLellan 2004); (Sidoli, Mantalaris and Asprey 2005)). If model parameters cannot be estimated from available data, the experimenter may invest in obtaining additional data or, alternatively, the inestimable parameters can be removed from the model or be fixed at some nominal value. Even if one is able to obtain additional data, the additional expense may not be justified if the new information does not significantly alter the model predictions in the region of interest. As a result, one is often interested in estimating only a subset of the unknown model parameters.

In this work, a first-principles based microalgal bioreactor model consisting of six ordinary differential equations and 12 unknown parameters is studied for estimability. It is shown that even under an optimal experimental design many of the process parameters do not have statistically significant effect on model predictions. Two model reparameterizing algorithms are proposed in order to reduce the number of parameters that must be estimated for accurate predictions. The first algorithm is based in a linear transformation of the parameter space while the second algorithm is based on a non-linear one. Both algorithms greatly decrease computational time while achieving a good degree of output prediction and significantly reducing computational complexity. The non-linear algorithm is superior to the linear one because it provides predictions which have a lower weighted sum of squared residuals and lower computational cost.

3.2 Microalgal bioreactor model

In an algal bioreactor system with an intracellular product, the two key states are the biomass (X) and intracellular product concentration (I_p). The intracellular product is algal oil which typically contains high-value substances such as very long chain polyunsaturated fatty acids (Drapcho, Nhuan and Walker 2008). The key nutrients controlling biomass production (growth rate) are carbon (C) and nitrogen (N). The biomass growth rate, μ, can be expressed as (Davidson and Cunningham 1996):

$$\mu = \mu_m f_1(N) f_2(C) \tag{3.1}$$

where f_1 and f_2 are functions of nitrogen and carbon respectively, and μ_m is the maximum growth rate. It has been shown that while the external carbon source concentration (S_2) can be directly related with the growth rate, this same assumption is not valid for nitrogen and other nutrients for algal systems (Tett and Droop 1988). Instead, the growth rate depends on the intracellular nitrogen concentration, or nitrogen quota (q) (Mailleret, Gouzé and Bernard 2005). In addition, there is a minimum cell quota (q_m) below which no growth is observed. Assuming general hyperbolic functions ((Davidson and Cunningham 1996); (Davidson 1996)) for the growth rate dependence on both S_2 and q, Equation 3.1 can be written as:

$$\mu = \mu_m \left(\frac{q - q_m}{K_q + q}\right)\left(\frac{S_2}{K_s + S_2}\right) \tag{3.2}$$

where K_q is the half saturation constant of nitrogen quota for growth and K_s is the half saturation constant of carbon source for growth. The cell quota is related to the external nutrient concentration by the rate at which cells can assimilate such nutrients. Several expressions have been proposed to model the uptake rate of nutrients by algae ((Caperon and Meyer 1972); (Tett and Droop 1988)). A simplified version of the Caperon and Meyer equation is used in this work, as it fits well the experimental data, and is computationally simpler:

$$\rho = \rho_m \left(1 - \frac{S_0}{S_1}\right)^{1+\varepsilon} \tag{3.3}$$

where ρ is the nitrogen uptake rate, S_1 is external nitrogen source concentration, ρ_m is the minimum cell quota for supporting growth, and S_0 can be interpreted as the concentration of nitrogen source at which the net uptake equals zero ((Passarge and Huisman 2002); (Caperon and Meyer 1972)). In Equation 3.3, ε is a small known perturbation factor introduced to remove the discontinuity in the first derivative of the expression. This perturbation factor does not significantly affect model predictions; however, it facilitates the solution of the differential equation system. Regression algorithms require the evaluation of the total prediction error at several points in the parameter space, which implies the solution of the system of differential equations. Therefore, the introduction of ε improve the performance of the of the parameter estimation, as there is a reduction in the required time to solve the system of differential equations.

Algal oil production, even if triggered by nitrogen deficiency conditions, is assumed to only depend on the carbon source concentration. In this case, a Michaelis-Menten relationship is assumed, with an additional saturation term that takes into account the decrease in algal oil production as cells become saturated with oil. The oil production rate, π, is given as follows:

$$\pi = \pi_m \frac{S_2}{K_\pi + S_2}\left(1 - \frac{I_p}{X}\right) \tag{3.4}$$

where π_m is maximum oil production rate and K_π is the half saturation constant for oil production. As oil is an intracellular product, and nitrogen can be stored without being converted into biomass, the total biomass can be expressed as the sum of three different compartments: oil storage (I_p), intracellular nutrient, and active biomass. The measurable biomass concentration (X) is therefore:

$$X = x + I_p + Q \tag{3.5}$$

where, x is the concentration of metabolically active biomass, and Q is the total concentration of intracellular nutrient in the reactor ($Q = qX$). The dynamics of the bioreactor are therefore described by a set of six differential equations shown in Table 3.1. Equations f_1-f_6 are first-order ODEs with the following six states:

1) x: functionally active biomass concentration [g/mL]
2) S_1: Nitrogen source concentration in culture media [g/mL]
3) S_2: Carbon source concentration in culture media [g/mL]
4) Q: Total nitrogen cellular quota [g/mL]
5) I_p: Total algal oil stored in cells [g/mL]
6) V: Total reaction volume [mL]

The process outputs are the total biomass concentration (X), the nitrogen source and carbon source concentration in culture media (S_1 and S_2, respectively), and the mass fraction of algal oil in the cell ($\frac{I_p}{X}$).

In this work, the bioreactor behaviour was investigated in fed-batch-mode (f_o = 0). To illustrate the numerical implementation of the proposed algorithm, nominal (i.e. "true") values of the system parameters were selected to lie within the range reported in literature, as shown in Table 3.2. Note that the proposed model reparameterization algorithms can be implemented for a different choice of nominal parameter values.

Table 3.1. Continuous bioreactor model

$$\frac{dx}{dt} = \mu x - x\frac{f_o}{V} - x(f_1^i + f_2^i - f_o)\frac{1}{V} \quad (f_1)$$

$$\frac{dS_1}{dt} = -\rho x + S_1^i\frac{f_1^i}{V} - S_1\frac{f_o}{V} - S_1(f_1^i + f_2^i - f_o)\frac{1}{V} \quad (f_2)$$

$$\frac{dS_2}{dt} = -\frac{1}{Y_{xs}}\mu x + S_2^i\frac{f_2^i}{V} - S_2\frac{f_o}{V} - k_m x - \frac{1}{Y_{ps}}\pi x - S_2(f_1^i + f_2^i - f_o)\frac{1}{V} \quad (f_3)$$

$$\frac{dQ}{dt} = \rho x - \frac{1}{Y_{xq}}\mu x - Q\frac{f_o}{V} - Q(f_1^i + f_2^i - f_o)\frac{1}{V} \quad (f_4)$$

$$\frac{dI_p}{dt} = \pi x - I_p\frac{f_o}{V} - I_p(f_1^i + f_2^i - f_o)\frac{1}{V} \quad (f_5)$$

$$\frac{dV}{dt} = f_1^i + f_2^i - f_o \quad (f_6)$$

System outputs

$$y(t) = \begin{bmatrix} X & S_1 & S_2 & \frac{I_p}{X} \end{bmatrix}^T \quad (y)$$

The initial guesses for parameter values differ by ten percent from the nominal values and are given in Table 3.3. The initial guesses are used in the experimental design and model reparameterization algorithms. The initial state conditions and other known quantities are also given in Table 3.3.

Table 3.2. Nominal parameter values

Parameter	Range	Nominal value	Unit	References
μ_m	0.01 – 0.23	0.15	1 / h	((Mendes, et al. 2007), (Droop 1974))
q_m	0.00275 – 0.032	0.027	g / g	((Gotham and Rhee 1981), (Caperon and Meyer 1972))
K_q	–	0.5 [a]	g / g	
ρ_m	0.0009 – 0.102	0.08	1 / h	((Davidson and Cunningham 1996), (Gotham and Rhee 1981))
K_s	0.0032 – 0.060	0.014	g / mL	((Wu, Shi and Shi 2007), (Shi, Zhang and Chen 2000))
s_o	0 – 3.5e-4	1.8e-5	g / mL	(Berland, et al. 1979)
Y_{xs}	0.12 – 0.68	0.56	g / g	(Patino, Janssen and von Stockar 2007)
k_m	0 – 0.005	0	g / g h	((Lee, Erickson and Yang 1985), (Shi, Zhang and Chen 2000))
Y_{ps}	–	0.4 [a]	g / g	
π_m	–	0.05 [a]	1 / h	
K_π	–	0.01 [a]	g / mL	
Y_{xq}	7.07 – 36.1	33.3	g / g	((Patino, Janssen and von Stockar 2007), (Chen and Johns 1991))

Notes: a. Parameter values were estimated from preliminary experimental data.

Table 3.3. Initial guess for parameter values, initial conditions and known quantities

Parameters				
Number	Parameter	Name	Initial guess	Unit
P1	μ_m	Maximum growth rate	0.135	1 / h
P2	q_m	Minimum cell quota for supporting growth	0.0243	g / g
P3	K_q	Half saturation constant of nitrogen quota for growth	0.55	g / g
P4	ρ_m	Maximum uptake rate	0.072	1 / h
P5	K_s	Half saturation constant of carbon source for growth	0.0154	g / mL
P6	s_o	Threshold substrate concentration	1.98e-5	g / mL
P7	$1/Y_{xs}$	Inverse of Biomass to substrate yield	1.98	g / g
P8	k_m	Maintenance constant	0	g / g h
P9	$1/Y_{ps}$	Inverse of product to substrate yield	2.25	g / g
P10	π_m	Maximum oil production rate	0.045	1 / h
P11	K_π	Half saturation constant for oil production	0.009	g / mL
P12	$1/Y_{xq}$	Inverse of biomass to substrate quota yield	0.033	g / g
Initial conditions				
1	x	Initial biomass concentration	4.275e-4	g / mL
2	S_1	Initial nitrogen source concentration	5e-4	g / mL
3	S_2	Initial carbon source concentration	0.04	g / mL
4	Q	Initial nitrogen cellular quota	2.25e-5	g / mL
5	I_p	Initial algal oil stored in cells	3e-4	g / mL
6	V	Initial reaction volume	1060	mL
Known quantities				
1	S_1^i	Nitrogen source concentration in inlet feed 1	0.01	g / mL
2	S_2^i	Carbon source concentration in inlet feed 2	0.2	g / mL

3.3 Experimental design

It is difficult to determine *a priori* which parameters have the greatest effect on the model predictions and hence can be estimated. For the purpose of this work, D-optimality criterion (Yao, et al. 2003) was chosen for input signal design.

Consider the following matrix whose entries are the partial derivatives of the state variables with respect to the parameters at specific times:

$$\tilde{Z} = \begin{bmatrix} \left(\frac{\partial z_1}{\partial P_1}\right)\Big|_{t_1} & \cdots & \left(\frac{\partial z_1}{\partial P_{12}}\right)\Big|_{t_1} \\ \vdots & & \\ \left(\frac{\partial z_6}{\partial P_1}\right)\Big|_{t_1} & \cdots & \left(\frac{\partial z_6}{\partial P_{12}}\right)\Big|_{t_1} \\ \vdots & & \\ \vdots & & \\ \left(\frac{\partial z_1}{\partial P_1}\right)\Big|_{t_f} & \cdots & \left(\frac{\partial z_1}{\partial P_{12}}\right)\Big|_{t_f} \\ \vdots & & \\ \left(\frac{\partial z_6}{\partial P_1}\right)\Big|_{t_f} & \cdots & \left(\frac{\partial z_6}{\partial P_{12}}\right)\Big|_{t_f} \end{bmatrix} \tag{3.6}$$

where the entry $\frac{\partial z_i}{\partial P_j}\Big|_{t_k}$ denotes the partial derivative of the i^{th} state (z) variable with respect to the j^{th} parameter at sample time t_k. The sampling time for this work was assumed to be one hour, as this is a realistic period for sampling and analysis.

The entries in the matrix $\tilde{Z}$ were computed integrating the sensitivity equations (Bard 1974) given by:

$$\frac{\mathrm{d}}{\mathrm{dt}}\left(\frac{\partial \mathrm{z}}{\partial \mathrm{P}}\right) = \frac{\partial f}{\partial \mathrm{z}} * \frac{\partial z}{\partial \mathrm{P}} + \frac{\partial f}{\partial \mathrm{P}} \tag{3.7}$$

where f is the vector values function whose entries are listed in Table 3.1. Equation 3.7 was integrating forward in time along with the model equations, given in Table 3.1, and computed at discrete intervals in order to obtain the entries of $\tilde{Z}$.

The matrix $\tilde{Z}$ describes the effect of parameter values on the system states at specific time instances $\mathrm{t} \in \{\mathrm{t}_1 \ldots \mathrm{t}_\mathrm{f}\}$. To determine the effect of parameters values on process outputs, which are measured quantities, the following relation is used:

$$\frac{\partial \mathrm{y}}{\partial \mathrm{P}} = \frac{\partial \mathrm{h}}{\partial \mathrm{P}} + \frac{\partial \mathrm{h}}{\partial \mathrm{z}}\frac{\partial \mathrm{z}}{\partial \mathrm{P}} \tag{3.8}$$

By computing the value of $\frac{\partial \mathrm{y}}{\partial \mathrm{P}}$ for $\mathrm{t} \in \{\mathrm{t}_1 \ldots \mathrm{t}_\mathrm{f}\}$ the output sensitivity matrix, Z, is defined as:

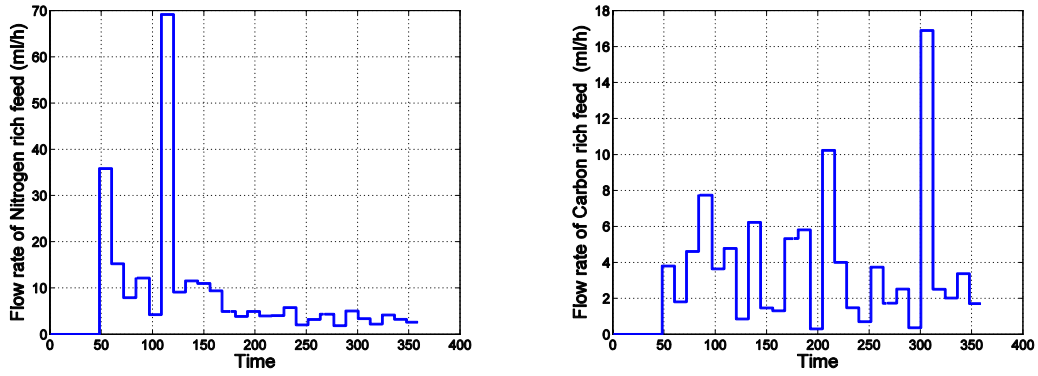
$$Z = \begin{bmatrix} \left(\frac{\partial y_1}{\partial P_1}\right)\Big|_{t_1} & \cdots & \left(\frac{\partial y_1}{\partial P_{12}}\right)\Big|_{t_1} \\ \vdots & & \\ \left(\frac{\partial y_4}{\partial P_1}\right)\Big|_{t_1} & \cdots & \left(\frac{\partial y_4}{\partial P_{12}}\right)\Big|_{t_1} \\ \vdots & & \\ \left(\frac{\partial y_1}{\partial P_1}\right)\Big|_{t_f} & \cdots & \left(\frac{\partial y_1}{\partial P_{12}}\right)\Big|_{t_f} \\ \vdots & & \\ \left(\frac{\partial y_4}{\partial P_1}\right)\Big|_{t_f} & \cdots & \left(\frac{\partial y_4}{\partial P_{12}}\right)\Big|_{t_f} \end{bmatrix} \tag{3.9}$$

The matrix Z describes the effect of parameter values on the measured outputs.

The optimal input was defined as one which maximizes the determinant of $(Z^{\mathrm{T}}Z)$. Mathematically, the optimal input, u^*, is computed as:

$$u^* = \arg\max_{u \in U} \left|\tilde{Z}^{\mathrm{T}}\tilde{Z}\right| \tag{3.10}$$

where *U* is the set of permissible inputs for the system. This class of permissible inputs was chosen as a set of piece-wise constant functions $u: t \mapsto R^2$defined on $t \in [48,360]$ (time units of hours), with a switch time of 12 hours. Note that the input was held at zero for the first 48 hours in order to accumulate an experimentally practical concentration of biomass in the reactor. The switch frequency of the input was chosen as 12 hours in order to maintain a balance between computational efficiency and process excitation. The total number of input changes (i.e. optimization variables) was 52. The final input specifications were chosen based on a computational time of 7 days. The total flow into the bioreactor is limited by the total reactor volume of 2.5L. This is taken into account by implementing a non-linear constraint in the optimization problem given on Equation 3.10. The input design was done using MATLAB® R2006b on an Intel Core2 Duo CPU with 2.39 GHz and 1 Gigabyte of RAM. The problem was setup as an optimization problem and solved using "Pattern search tool" available in MATLAB®. The designed input signals are shown in Figure 3.1.



(a) (b)

Figure 3.1. Designed input signals. (a) Flowrate of nitrogen rich feed. (b) Flowrate of carbon rich feed.

3.4 Parameter sensitivity analysis

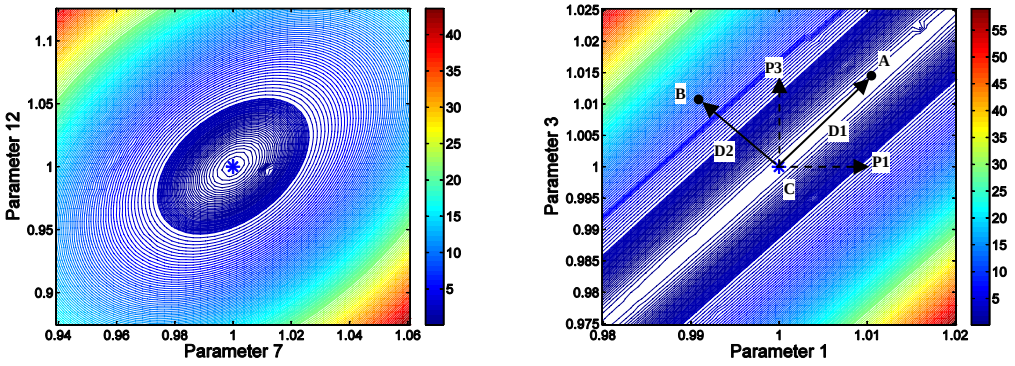
Even under the D-optimal experimental design computed in the previous section, not all parameters can be uniquely estimated using noisy measurements. More generally, it is often the case that not all model parameters have a strong effect on the model predictions (Yao, et al. 2003). To determine which parameters have a negligible effect on the model predictions, a perturbation-based test was performed to check for parameter estimability. Using the nominal parameter values, a nominal response was computed. Next, each of the system parameters was perturbed pair-wise. The model contains 12 parameters, and, as a result, a total of 66 graphs are computed. Laboratory measurements of X, S_1, S_2 and $\frac{I_p}{X}$ were assumed to have a measurement noise covariance of 1%, 1%, 1% and 5% respectively. Therefore, the weighing matrix, W, was chosen as:

$$W = \begin{bmatrix} 10000 & 0 & 0 & 0 \\ 0 & 10000 & 0 & 0 \\ 0 & 0 & 10000 & 0 \\ 0 & 0 & 0 & 400 \end{bmatrix} \tag{3.11}$$

whose diagonal entries are proportional to the inverse of the noise covariance of each output.

Contour plots of the weighted sum of squared error, WSSE, were obtained for each scenario. Two examples of these plots are shown in Figure 3.2. Figures 3.2a and 3.2b

shows the WSSE as a function of Parameters 12 ($1/Y_{xq}$) and 7 ($1/Y_{xs}$) and Parameters 3 (K_q) and 1 (μ_m), respectively. For ease of visualization the axis of the plots in Figure 3.2 were scaled by the nominal parameter values. Figure 3.2a shows that parameters 7 and 12 are estimable as there is significant change in WSSE in all directions. However, as shown in Figure 3.2b, the effects of some parameters on the model predictions are correlated. For example, consider points A, B and C in Figure 3.2b. The WSSE at points A and C is approximately equal whereas the WSSE is significantly different at points C and B. This suggests that there is a negligible change in WSSE along the directional vector, D1, whereas it changes significantly along D2. This implies that, in this case, neither Parameter 3 nor Parameter 1 is estimable. Rather, one linear combination of the two parameters is estimable. More generally, in the parameter space, there exist directions along which there is minimal change in the output predictions (e.g. the line D1 in Figure 3.2b). Conversely, there are directions along which the output prediction changes significantly (e.g. the line D2 in Figure 3.2b). Identification of these directions and subsequent reparameterization of the model can greatly reduce the computational time and complexity of parameter estimation and output prediction algorithms.



(a) (b)

Figure 3.2. Contour plots of the WSSE. (a) $1/Y_{xs}$ and $1/Y_{xq}$ (b) μ_m and K_q

3.5 Model reparameterization / Output Prediction

3.5.1 Linear algorithm

As shown in Figure 3.2b, there are linear combinations of parameters along which the WSSE will not change significantly. Ideally, one should estimate linear combinations of parameters that are estimable and fix inestimable parameter combinations at their nominal value. For parameter estimation, in the case of Figure 3.2b, the reparameterization could be achieved by defining a transformation on $\mathbb{R}^2$ from the basis {P1, P3} to a new basis {D1, D2}. After the transformation, only one variable, D2, needs to be estimated while the second variable, D1, can be fixed at its nominal value. In this way, the number of parameters to be estimated is reduced by one. For a system containing *n* parameters, one may define a transformation matrix, *H*, and a new set of parameters, *K*, given by:

$$K = H * P \tag{3.12}$$

where $H \in \mathbb{R}^{n*n}$, $P \in \mathbb{R}^n$ and $K \in \mathbb{R}^n$. The rows vectors in the *H* matrix transform the true parameter vector, *P*, to a pseudo-parameter vector, *K*. The main idea for model reparameterization is to choose *H* such that K_1 to K_d are estimable while K_{d+1} to K_n are inestimable, where $d \in \{1, 2,\ldots, n\}$. In order to identify the important directions (vectors) in the parameter space, the sensitivity matrix, Z, is used. To see how this is, define a parameter perturbation vector $h = (P-P_0)$ which lies in the null-space of *Z*. Consider the matrix *Y* containing all measured values given by:

$$Y = \left[y_1|_{t_1} \quad y_2|_{t_1} \quad \ldots \quad y_4|_{t_1} \quad y_1|_{t_2} \quad \ldots \quad y_4|_{t_2} \quad \ldots \quad y_1|_{t_f} \quad \ldots \quad y_4|_{t_f}\right] \tag{3.13}$$

about the nominal parameter value, P_o. Note that, by definition, $Z = \frac{\partial Y}{\partial P}\Big|_{P_0}$. The Taylor series expansion of *Y* about P_o is given by:

$$\begin{aligned} Y(P) &= Y(P_0) + \frac{\partial Y}{\partial P}\Big|_{P_0} * h + \frac{1}{2} h^T * \frac{\partial^2 Y}{\partial P^2}\Big|_{P_0} * h + \Delta(h) \\ &= Y(P_0) + Z * h + \frac{1}{2} h^T * \frac{\partial^2 Y}{\partial P^2}\Big|_{P_0} * h + \Delta(h) \end{aligned} \tag{3.14}$$

where $\Delta(h)$ represents the third and higher order terms in the Taylor series expansion. If h represents a sufficiently small perturbation (i.e. has a sufficiently small norm) then Equation 3.14 can be approximated by:

$$Y(P) = Y(P_0) + Z * h \tag{3.15}$$

If, furthermore, the vector h belongs to the null-space of Z then Equation 3.15 reduces to $Y(P) = Y(P_0)$ implying that no change has been observed in the measured values despite the perturbation in the parameters. Similarly, if h is such that the norm of $Z * h$ is small then the perturbation $h = (P\text{-}P_0)$ will have a small effect on measured values.

To compute the subspace of the parameter space which contains perturbation vectors with small effect on the measured variables, one can perform singular-value decomposition (SVD) (Chan 1982) on Z and identify its singular vectors which correspond to small singular values. This is because perturbing parameters along the direction given by a singular vector with a low corresponding singular value has little effect on the measured variables. To see how this is, consider s_i to be the singular value corresponding to the singular vector, h_i.

$$\lim_{s_i \to 0}(Z * h_i) = \lim_{s_i \to 0}(s_i * h_i) = 0 \tag{3.16}$$

If s_i is small then so is the norm of $Z * h_i$ and according to Equation 3.15 this implies that a perturbation along h will have only a small effect on the measured variables.

Similarly, varying parameters along the direction given by a singular vector corresponding to a high singular value will have a large effect on the model predictions. This property can be used to partition the parameter space into orthogonal linear subspaces corresponding to estimable and inestimable parameter combinations. Therefore, the actual parameter space can be transformed into a pseudo-parameter space using Equation 3.12 and a matrix H whose rows are the singular vectors of Z. The rows (i.e. singular vectors) in H are arranged in descending order of the corresponding singular values. Therefore, vector h_1 in Equation 3.15 has a corresponding singular value s_1. Similarly, h_2 corresponds to s_2 and so on with $s_1 > s_2 > s_3 > \cdots > s_{12}$.

$$h = [h_1 \quad h_2 \quad \cdots \quad h_{12}] \qquad s = \begin{bmatrix} s_1 & 0 & \cdots & 0 \\ 0 & s_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & 0 \\ 0 & 0 & 0 & s_{12} \end{bmatrix} \tag{3.17}$$

Under the proposed formulation only a subset of the elements of the transformed (or pseudo-) parameters *K* should be estimated. Specifically, the elements of *K* corresponding to low singular values can be fixed at a nominal value because they have little effect on the measured variables and therefore little effect on the model WSSE (a measure of the output prediction). Conversely, the elements of *K* corresponding to large singular values are optimization variables in the output prediction algorithm. Note that the transformation in Equation 3.12 is linear, and, as a result, it is expected to be valid locally about initial guess parameter values used in experimental design. For the bioreactor model, the first four *K* values (i.e. $\{K_1 \dots K_4\}$) were chosen after visual comparison of the predicted output and a nominal response based on nominal parameter values (in Table 3.3). Four was the minimum number of parameter which could be used to match the nominal model predictions.

The optimization problem was formulated as follows:

$$[K_1 \dots K_4] = \underset{K_1 \dots K_4}{\operatorname{argmin}} \sum_{t=1}^{t_f} (y_t - \tilde{y}_t)^2 \tag{3.18}$$

where $\tilde{y}$ is the predicted output based on $K_1 \dots K_4$ and t_f is the final time.

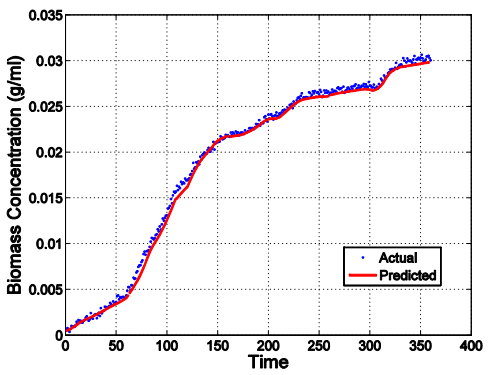
To highlight the ability of the simplified model (containing only four free parameters) to match the predictions of the 12 parameter model the following procedure was implemented. First, a simulation was performed using the nominal set of parameters (in Table 3.3) to obtain a set of observations. Next, Gaussian noise with covariance of 1%, 1%, 1% and 5% for outputs X, S_1, S_2 and $\frac{I_p}{X}$, respectively, was added to the observations. Using these noise corrupted observations the pseudo-parameter $K_1 \dots K_4$ were estimated. Finally, the model predictions based on the estimated values for $K_1 \dots K_4$ were compared to the noise corrupted observations. Note that the input signal used to generate the data

was the D-optimal signal, shown in Figure 3.1 and that 360 total observations were used for parameter estimation.

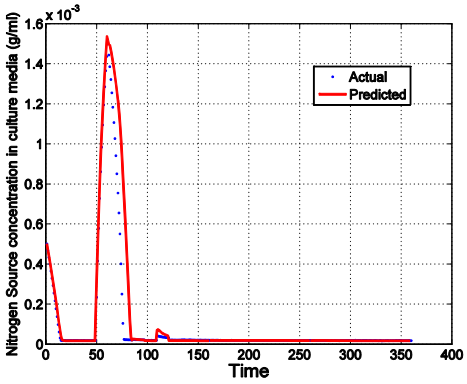
The results of parameter estimation from the four-parameter model as well as the output prediction are shown in Table 3.4 and Figure 3.3 respectively. As shown in Figure 3.3, the predictions obtained by using the simplified (four-parameter) model follow the general trend of the outputs given by the true model. As shown in Table 3.4, however, this does not imply that all of the original parameters (i.e. *P*) were accurately estimated. Indeed, as expected, many of the system parameters are inestimable even using data from an optimal experiment design. The benefits of using the four-parameter model are three-fold. First, the inestimable parameters are computed and known explicitly as opposed to being selected implicitly by the optimizer. Secondly, estimating only four parameters is far more computationally efficient than estimating twelve parameters. Finally, the method is able to achieve good output prediction even under an input which is optimally designed to identify all twelve original parameters.

Table 3.4. Results of parameter estimation (linear algorithm)

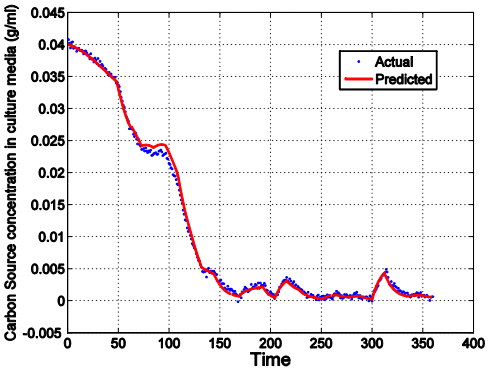
#	Parameter	Actual value	Estimated value	Error (%)	#	Parameter	Actual value	Estimated value	Error (%)
1	μ_m	0.15	0.1347	10.2	7	$1/Y_{xs}$	1.8	1.8076	0.42
2	q_m	0.027	0.0249	7.6	8	k_m	0	0	0
3	K_q	0.5	0.5507	10.1	9	$1/Y_{ps}$	2.5	2.4952	0.19
4	ρ_m	0.08	0.0713	10.9	10	π_m	0.05	0.0642	28.3
5	K_s	0.014	0.0099	29.6	11	K_π	0.01	0.0079	20.7
6	s_o	1.8e-5	1.798e-5	0.08	12	$1/Y_{xq}$	0.03	0.0296	1.3



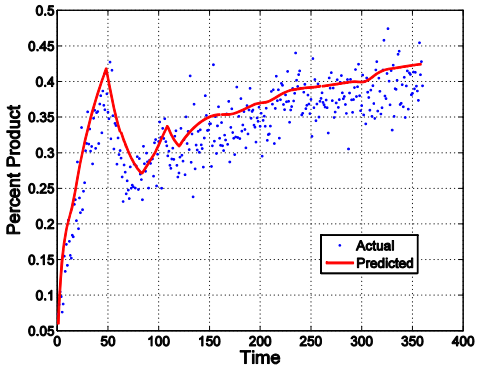
(a)



(b)



(c)



(d)

Figure 3.3. Comparison of predicted and actual outputs (linear algorithm). (a) $y_1 = X$ (b) $y_2 = S_1$ (c) $y_3 = S_2$ (d) $y_4 = \frac{I_p}{X}$

3.5.2 Non-linear algorithm

The transformation in Equation 3.12 is linear and is expected to be valid locally about initial guess parameter values used to derive the linear transformation in the parameter space. In order to obtain a transformation that is valid over a wider range in the parameter space, a transformation-based approach for reparameterizing unidentifiable or inestimable systems is investigated (Ben-Zvi 2008). This approach applies differential geometry and *a priori* knowledge of the model structure in the parameter estimation. The method reparameterizes the non-linear ODE system by indentifying process quantities (such as reaction rates) that have a significant impact on model behaviour. The central idea in this algorithm is to accurately estimate these process quantities, labelled pseudo-outputs, even if each specific model parameter is not accurately estimable. Choosing of the pseudo-outputs partitions the parameter space into estimable and inestimable subspaces. This approach is not sensitive to experimental conditions and does not require sensitivity information. The method proposed in (Ben-Zvi 2008) is implemented in this work with some necessary modifications.

The pseudo-outputs are usually non-linear combinations of parameters or states. For this study, the pseudo-outputs are chosen to be the three reaction rates (μ, ρ and π) given in Table 3.1. $\phi^+(P)$ is the set of pseudo-outputs and is given in Equation 3.19.

$$\phi^{+}(P) = \left[\begin{array}{c} \mu_m * \dfrac{\dfrac{Q}{X} - q_m}{K_q + \dfrac{Q}{X}} * \dfrac{S_2}{K_s + S_2} \\ \\ \rho_m \left(1 - \dfrac{S_o}{S_1}\right)^{1.001} \\ \\ \pi_m \dfrac{S_2}{K_\pi + S_2} \left(1 - \dfrac{I_p}{X}\right) \end{array} \right]_{x=x_0, S_1=S_{1_0}, S_2=S_{2_0}, Q=Q_0, I_p=I_{p_0}, V=V_0} \tag{3.19}$$

where $x_0, S_{1_0}, S_{2_0}, Q_0, I_{p_0}$ and V_0 are initial conditions of the system states, given on Table 3.3. Once $\phi^{+}(P)$ is chosen, a set of parameter combinations $\phi^{-}(P)$ must be found such that the mapping $\Phi = [\phi^{+}(P), \phi^{-}(P)]$ is a local diffeomorphism, "on an open and dense subset of the parameter space" (Ben-Zvi 2008), and satisfies the following Equation 3.20.

$$\langle \frac{\partial \phi_i^{+}}{\partial P}, \frac{\partial \phi_j^{-}}{\partial P} \rangle = 0 \tag{3.20}$$

for all $i \in \{1,2,\ldots d_1\}$ and $j \in \{1,2,\ldots d\text{-}d_1\}$, where d and d_1 are the total number of system parameters and pseudo-outputs respectively. In this case, d is twelve and d_1 is three. Let M and N be differentiable spaces. A function, $f : M \to N$ is a local diffeomorphism, if every point m in M, there exists an open set W containing m, such that $f(W)$ is open in N and $f|_W : W \to f(W)$ is a diffeomorphism. A diffeomorphism is an invertible and differentiable function that maps one differentiable space to another.

Equation 3.20 ensures $\phi^{+}$and ϕ^{-} are independent, that is, the estimates for $\phi^{+}$are independent of the fixed values used for ϕ^{-}. This is critical as values obtained for ϕ^{+} are not implicit functions of the values of ϕ^{-} (Ben-Zvi 2008). The calculation of $\phi^{-}(P)$ becomes complicated for systems with more than a few parameters. However, the method of characteristics (John 1971) can be used to find a set of coordinates, $\phi^{-}(P)$, that are orthogonal to a known set of coordinates, $\phi^{+}(P)$. Under this method, finding $\phi^{-}(P)$ amounts to finding functions, $\phi_1^{-}, \phi_2^{-}, \ldots, \phi_{d-d_1}^{-}$, such that for each function ϕ_j^{-} with $j \in \{1,2,\ldots,d\text{-}d_1\}$, and all $i \in \{1,2,\ldots,d_1\}$, Equation 3.20 holds. Evaluating Equation 3.20 over all $i \in \{1,2,\ldots,d_1\}$ implies that the following equations hold for all $j \in \{1,2,\ldots,d\text{-}d_1\}$:

$$\sum_{l=1}^{d} f_l^1(P)\frac{\partial \phi_j^-}{\partial P_l} = 0$$
$$\sum_{l=1}^{d} f_l^2(P)\frac{\partial \phi_j^-}{\partial P_l} = 0 \qquad (3.21)$$
$$\vdots$$
$$\sum_{l=1}^{d} f_l^{d_1}(P)\frac{\partial \phi_j^-}{\partial P_l} = 0$$

where $f_l^i(P) = \frac{\partial \phi_i^+}{\partial P_l}(P)$. Equation 3.21 is a set of linear partial differential equations. The gradient $f_l^i(P)$ for the bioreactor system, with $\phi^+(P)$ as given in Equation 3.19, is given in Table 3.5.

Table 3.5. $f_l^i(P)$ functions

		l											
		1	2	3	4	5	6	7	8	9	10	11	12
i	1	$\frac{(Q-P2)*S_2}{(P3+Q-P2)(1+S_2)}$	$-\frac{P1*S_2*P3}{(P3+Q-P2)^2(1+S_2)}$	$-\frac{P1*(Q-P2)*S_2}{(P3+Q-P2)^2(1+S_2)}$	0	0	0	0	0	0	0	0	0
	2	0	0	0	$\frac{S_1-P6}{P5+S_1-P6}$	$\frac{P4*(P6-S_1)}{(P5+S_1-P6)^2}$	$\frac{P4*P5}{(P5+S_1-P6)^2}$	0	0	0	0	0	0
	3	0	0	0	0	0	0	0	0	0	$\frac{S_2}{P11+S_2}$	$\frac{-P10*S_2}{(P11+S_2)^2}$	0

In order to solve Equation 3.21 using the method of characteristics a set of nonlinear ODEs given by $\dot{\xi} = \frac{\partial \phi^+}{\partial P}\Big|_{P=\xi}$ must be analytically solved. In general, a set of non-linear ODEs do not have an analytical solution (John 1971). Using MAPLE™ 11.0, no analytical solution was obtained. As a result, the method proposed by Dr. Ben-Zvi (2008) was modified by approximating $\phi^+(P)$ by a second-order Taylor series expansion, about the initial guesses for parameters, given by:

$$\tilde{\phi}^{+}(P) = \phi^{+}(P_0) + \frac{\partial \phi^{+}}{\partial P}\bigg|_{P_0} * (P - P_o) + \frac{1}{2}(P - P_o)^T * \frac{\partial^2 \phi^{+}}{\partial P^2}\bigg|_{P_0} * (P - P_o) \qquad (3.22)$$

where $\tilde{\phi}^{+}(P)$ is the approximation of $\phi^{+}(P_0)$.

The advantage of using a second-order Taylor series approximation for $\phi^{+}(P_0)$ is that the gradient given by:

$$\tilde{f}^{i}(P) = \frac{\partial \tilde{\phi}_i^{+}}{\partial P}(P) \qquad (3.23)$$

is linear and therefore the differential equation:

$$\dot{\xi} = \frac{\partial \tilde{\phi}^{+}}{\partial P}\bigg|_{P=\xi} \qquad (3.24)$$

is a set of linear time-invariant ordinary differential equations (LTI ODE) which has an analytic solution given by the variation of parameters formulation (Kohler and Johnson 2006). For the bioreactor system with the $\tilde{f}^{i}(P) = [\tilde{f}_1^{i}(P) \quad \ldots \quad \tilde{f}_{12}^{i}(P)]$ as defined in Equation 3.23, the values of $\tilde{f}_l^{i}(P)$ for $l = 1,2,\ldots,12$ are given by:

$$\tilde{f}_l^{i}(P) = A_i + B_i * P \qquad (3.25)$$

The A_i vectors and B_i matrices are given by:

$$A_1 = \begin{bmatrix} -0.00177 \\ -0.0103 \\ 0.000456 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \qquad B_{1=} \begin{bmatrix} 0 & -0.0765 & 0.00338 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -0.0765 & -0.0393 & 0.0205 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0.00338 & 0.0205 & -0.00173 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$A_2 = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0.06080 \\ -0.271 \\ -4.120 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \qquad B_2 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1.904 & -61.067 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1.904 & 17.267 & 268.2425 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -61.067 & 268.242 & -553.752 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$A_3 = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0.966 \\ -0.275 \\ 0 \end{bmatrix} \qquad B_3 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -16.660 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -16.660 & 30.599 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Following the procedure outline in (Ben-Zvi 2008), nine independent vectors ($v_1, v_2, \ldots, v_9$) are chosen such that the span$\{\tilde{f}_l^1, \tilde{f}_l^2, \tilde{f}_l^3, v_1, v_2, \ldots, v_9\} = R^{12}$. The following $v_1, v_2, \ldots, v_9$ are chosen in this case:

$$\begin{aligned} v1 &= [1 \; 0 \; 0 \; 0 \; 0 \; 0 \; 0 \; 0 \; 0 \; 0 \; 0 \; 0] \\ v2 &= [0 \; 1 \; 0 \; 0 \; 0 \; 0 \; 0 \; 0 \; 0 \; 0 \; 0 \; 0] \\ v3 &= [0 \; 0 \; 0 \; 1 \; 0 \; 0 \; 0 \; 0 \; 0 \; 0 \; 0 \; 0] \\ v4 &= [0 \; 0 \; 0 \; 0 \; 1 \; 0 \; 0 \; 0 \; 0 \; 0 \; 0 \; 0] \\ v5 &= [0 \; 0 \; 0 \; 0 \; 0 \; 0 \; 1 \; 0 \; 0 \; 0 \; 0 \; 0] \\ v6 &= [0 \; 0 \; 0 \; 0 \; 0 \; 0 \; 0 \; 1 \; 0 \; 0 \; 0 \; 0] \\ v7 &= [0 \; 0 \; 0 \; 0 \; 0 \; 0 \; 0 \; 0 \; 1 \; 0 \; 0 \; 0] \\ v8 &= [0 \; 0 \; 0 \; 0 \; 0 \; 0 \; 0 \; 0 \; 0 \; 0 \; 1 \; 0] \\ v9 &= [0 \; 0 \; 0 \; 0 \; 0 \; 0 \; 0 \; 0 \; 0 \; 0 \; 0 \; 1] \end{aligned} \tag{3.26}$$

The next step involves computing the solution of the differential equations specified by $\dot{\xi} = \tilde{f}_l^1, \dot{\xi} = \tilde{f}_l^2$ and $\dot{\xi} = \tilde{f}_l^3$, as a function of a time-like variable τ_1, τ_2 and τ_3 respectively (Ben-Zvi 2008). The mathematical expressions of these solutions are very long and complex and therefore are not shown here. Similarly, the solutions of the differential

equations given by $\dot{\xi} = v1$, $\dot{\xi} = v2$, …, $\dot{\xi} = v9$ are computed as a function of τ_4, τ_5, …, τ_{12} respectively. The solutions are shown below.

$$\xi(\tau_4) = \begin{bmatrix} \xi_1(0) + \tau_4 \\ \xi_2(0) \\ \xi_3(0) \\ \vdots \\ \xi_{12}(0) \end{bmatrix} \qquad \xi(\tau_5) = \begin{bmatrix} \xi_1(0) \\ \xi_2(0) + \tau_5 \\ \xi_3(0) \\ \vdots \\ \xi_{12}(0) \end{bmatrix} \qquad \xi(\tau_6) = \begin{bmatrix} \xi_1(0) \\ \vdots \\ \xi_4(0) + \tau_6 \\ \vdots \\ \xi_{12}(0) \end{bmatrix}$$

$$\xi(\tau_7) = \begin{bmatrix} \xi_1(0) \\ \vdots \\ \xi_5(0) + \tau_7 \\ \vdots \\ \xi_{12}(0) \end{bmatrix} \qquad \xi(\tau_8) = \begin{bmatrix} \xi_1(0) \\ \vdots \\ \xi_7(0) + \tau_8 \\ \vdots \\ \xi_{12}(0) \end{bmatrix} \qquad \xi(\tau_9) = \begin{bmatrix} \xi_1(0) \\ \vdots \\ \xi_8(0) + \tau_9 \\ \vdots \\ \xi_{12}(0) \end{bmatrix}$$

$$\xi(\tau_{10}) = \begin{bmatrix} \xi_1(0) \\ \vdots \\ \xi_9(0) + \tau_{10} \\ \vdots \\ \xi_{12}(0) \end{bmatrix} \qquad \xi(\tau_{11}) = \begin{bmatrix} \xi_1(0) \\ \vdots \\ \xi_{10}(0) \\ \xi_{11}(0) + \tau_{11} \\ \xi_{12}(0) \end{bmatrix} \qquad \xi(\tau_{12}) = \begin{bmatrix} \xi_1(0) \\ \vdots \\ \xi_{10}(0) \\ \xi_{11}(0) \\ \xi_{12}(0) + \tau_{12} \end{bmatrix}$$

A mapping M is defined as a compositions of all the solutions $\xi(\tau_1)$ to $\xi(\tau_{12})$:

$$M(\tau) = \begin{bmatrix} \xi_1(\tau_1, \tau_2, \tau_3, \dots, \tau_{12}) \\ \xi_2(\tau_1, \tau_2, \tau_3, \dots, \tau_{12}) \\ \xi_3(\tau_1, \tau_2, \tau_3, \dots, \tau_{12}) \\ \vdots \\ \xi_{12}(\tau_1, \tau_2, \tau_3, \dots, \tau_{12}) \end{bmatrix} = \xi(\tau_1) \circ \cdots \circ \xi(\tau_{12}) \tag{3.27}$$

evaluated at $\xi_1(0) = \xi_2(0) = \cdots = \xi_{12}(0) = 0$. The partial derivatives, $\left[\frac{\partial}{\partial P_1}, \frac{\partial}{\partial P_2}, \frac{\partial}{\partial P_3}, \dots, \frac{\partial}{\partial P_{12}}\right]$, are the basis with respect to which the coordinates ξ are defined. Therefore, the substitutions $\xi_1 \to P1, \xi_2 \to P2, \dots, \xi_{12} \to P12$ can be made which results in the following equation:

$$P = M(\tau) \tag{3.28}$$

where the mapping M is given by:

$$M(\tau) = \begin{bmatrix} -2.089 * 10^8 + 0.0537e^{-0.102\tau_1} + 2.089 * 10^8 e^{-4.821*10^{-13}\tau_1} + 0.0625e^{0.0610\tau_1} + \tau_4 \\ -0.376 * 10^8 + 0.0709e^{-0.102\tau_1} + 0.376 * 10^8 e^{-4.821*10^{-13}\tau_1} - 0.0504e^{0.0610\tau_1} + \tau_5 \\ -8.513 * 10^8 - 0.0164e^{-0.102\tau_1} + 8.513 * 10^8 e^{-4.821*10^{-13}\tau_1} - 0.0131e^{0.0610\tau_1} \\ 4.108 * 10^7 + 0.450 * 10^{-3} e^{-664.730\tau_2} - 4.108 * 10^7 e^{-7.038*10^{-10}\tau_2} + 0.00272e^{128.245\tau_2} + \tau_6 \\ 8.787 * 10^6 - 0.186 * 10^{-2} e^{-664.730\tau_2} - 8.787 * 10^6 e^{-7.038*10^{-10}\tau_2} - 0.0129e^{128.245\tau_2} + \tau_7 \\ -2.740 * 10^5 + 0.475 * 10^{-2} e^{-664.730\tau_2} + 2.740 * 10^5 e^{-7.038*10^{-10}\tau_2} - 0.00531e^{128.245\tau_2} \\ -5.940 + \tau_8 \\ \tau_9 \\ -6.750 + \tau_{10} \\ -2.0 * 10^{-11} + 1.760 * 10^{-40} e^{15.300\tau_3}(-7.668 * 10^{38} \cosh(22.619\tau_3) + 7.990 * 10^{38} \sinh(22.619\tau_3)) \\ 0.040 + 5.634 * 10^{-39} e^{15.300\tau_3}(-1.189 * 10^{37} \cosh(22.619\tau_3) + 9.605 * 10^{36} \sinh(22.619\tau_3)) + \tau_{11} \\ -0.099 + \tau_{12} \end{bmatrix} \quad (3.29)$$

Equation 3.28, gives a transformation from the pseudo-parameter vector, τ, to the actual parameter vector, *P*. The computation of the pseudo-parameter vector $\tau = M^{-1}(P)$ could not be done analytically. It was therefore computed numerically for specific values of *P*. Note that the inverse transformation needs to be solved only once in order to obtain the nominal values of τ corresponding to the initial guesses for *P*, given in Table 3.3. Once the nominal values of τ are obtained, [τ_4, τ_5, ..., τ_{12}] can be fixed at their nominal values, while [τ_1, τ_2, τ_3] become the optimization variables for output prediction. Thus, reducing the number of optimization variables from 12 to 3. A summary of the output prediction algorithm using the non-linear transformation is given in the flowchart in Figure 3.4.

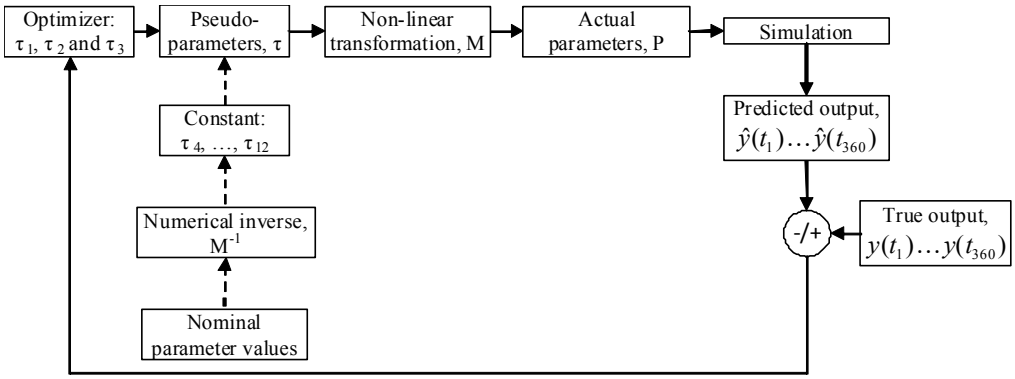


Figure 3.4. A flowchart of the parameter estimation algorithm using the non-linear transformation.

Figure 3.4 contains both dashed and solid lines. The dashed lines denote the steps that are only done once in the beginning of the algorithm. The solid lines in Figure 3.4 denote steps which are done iteratively. Note that both the numerical inversion of *M* and integration of Equation 3.24 (used to obtain *M*) need only be done once.

To test the ability of the simplified three-parameter model to match the full-model predictions, the procedure used to evaluate the four-parameter (linear reparameterization based) model proposed in Section 5.1 was repeated. First, a simulation was performed using the nominal set of parameters (in Table 3.3) to obtain a set of observations. Next, Gaussian noise with covariance of 1%, 1%, 1% and 5% for outputs X, S_1, S_2 and $\frac{I_p}{X}$, respectively, was added to the observations. Using these noise corrupted observations the pseudo-parameter $\tau_1 \dots \tau_3$ were estimated. Model predictions based on the estimated values for $\tau_1 \dots \tau_3$ were compared to the noise corrupted observations. As before, the input signal used to generate the data was the D-optimal signal shown in Figure 3.1, and 360 total observations were used for parameter estimation.

The results of parameter estimation from the three-parameter model as well as the output prediction are shown in Table 3.6 and Figure 3.5 respectively. As shown in Figure 3.5, the predictions obtained by using the simplified (three-parameter) model match the predictions made by the full model. However, this does not imply that all parameters are accurately estimated, as seen on Table 3.6. Similar to the linear case, several system parameters are inestimable. There are two key advantages of using the non-linear algorithm when compare to its linear counterpart. The variables required to optimize for output prediction are reduced from 4 to 3. The reduction in the number of parameters, combined with the non-linear transformation, resulted in the reduction in the computational time of 49%. The computation time and WSSE for both the linear and non-linear algorithms are shown in Table 3.7. A second advantage of the non-linear algorithm is that it achieved better output predictions that the linear algorithm especially for carbon source concentration (S_2) and mass fraction of algal oil in the cell $\left(\frac{I_p}{X}\right)$.

Table 3.6. Results of parameter estimation (non-linear algorithm)

Number	Parameter	Actual value	Estimated value	Error (%)	Number	Parameter	Actual value	Estimated value	Error (%)
1	μ_m	0.15	0.1678	11.9	7	Y_{sx}	1.8	1.98	10
2	q_m	0.027	0.029	7.4	8	k_m	0	0	0
3	K_q	0.5	0.5305	6.1	9	Y_{sp}	2.5	2.25	10
4	ρ_m	0.08	0.0744	7.0	10	π_m	0.05	0.0497	0.6545
5	K_s	0.014	0.011	21.1	11	K_π	0.01	0.0082	18.3
6	s_o	1.8e-5	2.17e-5	20.6	12	Y_{qx}	0.03	0.033	10

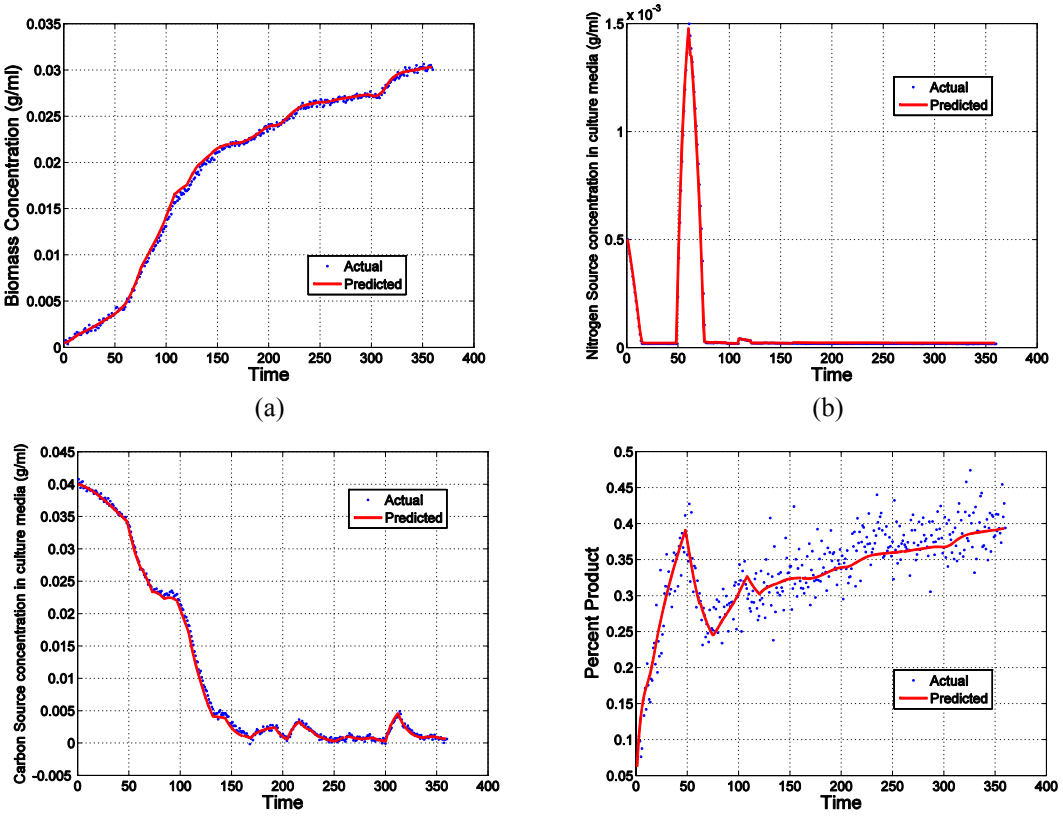


Figure 3.5. Comparison of predicted and actual outputs (non-linear algorithm). (a) $y_1 = X$ (b) $y_2 = S_1$ (c) $y_3 = S_2$ (d) $y_4 = \frac{I_p}{X}$

3.5.3 Comparison of the linear and non-linear algorithm

One way to compare the two algorithms is by examining their resulting weighted sum of squared error, WSSE. For the sake of comparing the linear and non-linear algorithms, the WSSE is calculated for four different scenarios. In Scenario 1, the WSSE is calculated using the nominal ("true") parameter values, to predict output. This gives a measure of noise-level present in the system. In Scenario 2, output prediction is done by estimating all 12 parameter of the bioreactor model given in Table 3.1. The nominal parameter values are used as initial guesses for parameter estimation. This gives an idea of the best level of output prediction that is possible. The four-parameter model, obtained from using the linear algorithm for model reparameterization, is used in Scenario 3. Subsequently, Scenario 4 utilizes the three-parameter model obtained using the non-linear reparameterization algorithm. Table 3.7 contains the WSSE values and computational time for each of the four scenarios. From Table 3.7 it can be seen that the non-linear

algorithm's WSSE is significantly lower that the linear one, as a result it is better for output prediction. Furthermore, the WSSE of the three-parameter model is comparable to the WSSE of the 12-parameter model, which is computationally more complex and takes more time to converge even when the nominal parameter values are given as initial guesses. This implies that the nine parameter combinations not estimated in the non-linear algorithm are indeed inestimable. Furthermore, if the nominal parameter values were not used as the initial guesses, then output prediction using 12-parameter model faces two major obstacles. Firstly, the time for the algorithm to converge significantly increases. Secondly, the result obtained could be a local minimum. In this aspect, the chances of a result from the non-linear algorithm being an only local minimum are lower, as it only utilizes three variables in its optimizer.

Table 3.7. Weighted sum of squared error, WSSE, for the four scenarios

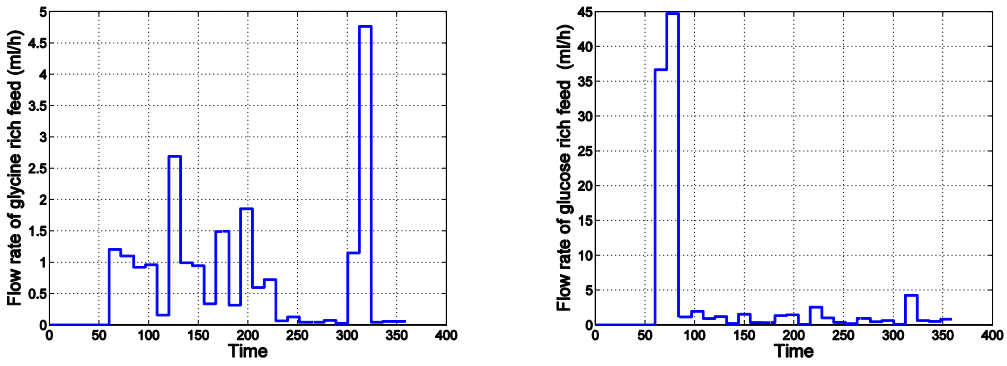
Scenario	Description	WSSE	Computational time
1	Nominal parameters values	108.1	-
2	12 parameter model	107.9	19 hours and 32 minutes
3	Four-parameter model (linear algorithm)	227.9	2 hours and 46 minutes
4	Three-parameter model (non-linear algorithm)	112.4	1 hours and 06 minutes

3.6 Experimental design based on pseudo-parameters

The D-optimal experimental design in Section 3 is based on maximizing the determinant of $(Z^{\mathrm{T}}Z)$ which is optimal for the estimation of all twelve original parameters (P). This design is not optimal with respect to identifying the three pseudo-parameters (τ_1, τ_2 and τ_3) in the non-linear parameter estimation algorithm. As a practical matter, one would like to identify only the estimable parameters. As a result, it is necessary to compute an input that is optimal for the estimation of the three pseudo-parameters. Therefore, the objective function of the D-optimal design may be modified to maximize the determinant of $(\hat{Z}^T\hat{Z})$, where $\hat{Z} = \frac{\partial Y}{\partial \tilde{\tau}}\Big|_{\tilde{\tau}=M^{-1}(P_o)}$ and $\tilde{\tau} = [\tau_1, \tau_2, \tau_3]$. $\hat{Z}$ can be calculated as follows:

$$\hat{Z} = \frac{\partial Y}{\partial \tilde{\tau}}\Big|_{\tilde{\tau}=M^{-1}(P_o)} = \mathrm{Z} * \frac{\partial M}{\partial \tilde{\tau}}\Big|_{\tilde{\tau}=M^{-1}(P_o)} \tag{3.30}$$

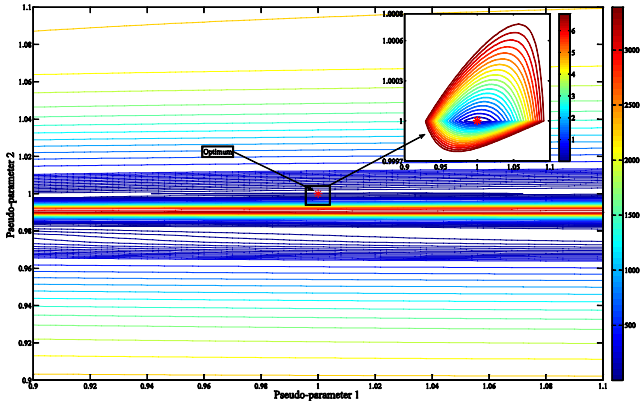
where $\frac{\partial M}{\partial \tilde{\tau}}$ can be calculated using Equation 3.28. As with the 12 parameter experimental design, the function space over which the input is optimized is the set of piecewise inputs with a switch frequency of 12 hours. The signal is design for an experiment of 15 days with the first two days being zero input. The designed input signals are shown in Figure 3.6. The input signal shown in Figure 3.6 can be used to obtain estimates of $\tilde{\tau}$ by implementing the design using a bioreactor system.



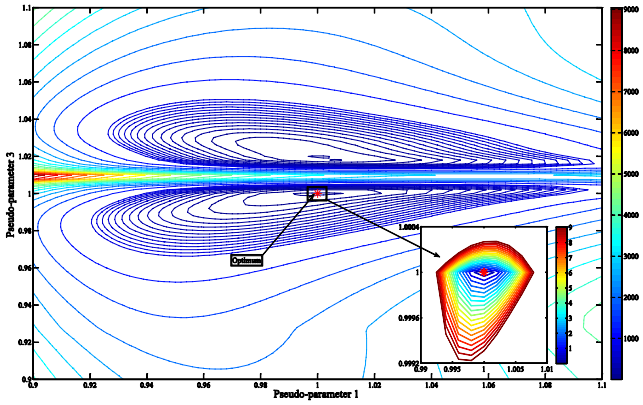
(a) (b)

Figure 3.6. Designed input signals (optimal with respect to pseudo-parameters). (a) Flowrate of nitrogen rich feed. (b) Flowrate of carbon rich feed.

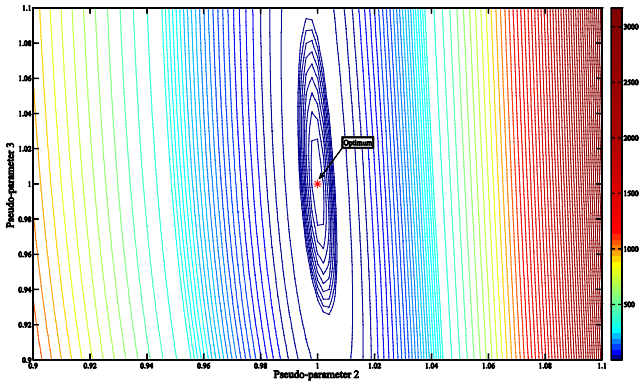
The input signal shown in Figure 3.6 was used to examine the estimability of the pseudo-parameters using the perturbation-based approach discussed in Section 4. There are three pseudo-parameters; as a result, three contour plots of the WSSE, as a function of the pseudo-parameter values, were obtained and plotted in Figure 3.7. For ease of comparison, the axes of the plots in Figure 3.7 were scaled to the nominal value of the pseudo-parameters. Figure 3.7 shows that pseudo-parameters 1, 2 and 3 are estimable because the WSSE exhibits a local minimum at the true pseudo-parameter values. The existence of this local minimum is visualized as closed contours around the minimum in Figure 3.7. It is important to note that other local minima are observed in Figures 7a and 7b. Therefore, during the parameter estimation algorithm, a global optimizer should be used to obtain the best parameter estimates.



(a)



(b)



(c)

Figure 3.7. Contour plots of the WSSE (for the pseudo-parameters). (a) Pseudo-parameter 1 and 2 (b) Pseudo-parameter 1 and 3 (c) Pseudo-parameter 2 and 3

3.7 Conclusions

Microalgae are efficient biological systems with a higher biomass production and faster growth rate than other energy crops ((Minowa, et al. 1995); (Miao and Wu 2006)). Moreover, production of microalgae does not require the use of crop area, as it can be grown in desertic areas or intensively in bioreactors. Microalgae have the ability to produce and store up to 80% of their dry weight as oil that can be used for the production of biodiesel (Drapcho, Nhuan and Walker 2008).

An ODE model for a microalgal bioreactor system is proposed. This first-principles based model contains twelve unknown parameters and six states. The model considers biomass, two substrates, intracellular nitrogen concentrations, as well as intracellular product. Estimability of the reactor model is investigated in this work. To determine which parameters have a negligible effect on the model predictions, a nominal output response was computed and compared to an output obtained by perturbing each of the system parameters pair-wise. It was found that not all parameters had unique contributions to model predictions. This conclusion was true even when a D-optimal input signal was used to excite the system.

A set of eight linear combinations of parameters were identified, along which minimal change in output occurs, using singular value decomposition of the parameter covariance matrix. Using this fact, the linear model reparameterization was used to partition the parameter space into estimable and inestimable linear subspaces. As a result, the number of free unknown quantities (i.e. pseudo-parameters) in the model was reduced from twelve to four. The linear model reparameterization algorithm greatly decreased the computational expense of the parameter estimation problem while achieving a good degree of output prediction.

A non-linear algorithm was also used for model reparameterization. In this case, the three system rate functions were chosen as pseudo-outputs. Method of characteristics was used to determine the inestimable parameter combinations. A non-linear transformation was used to partition the parameter space into estimable and inestimable subspaces. In the non-linear case, only three pseudo-parameters needed to be estimated in order to match the output predictions of the nominal model. This non-linear algorithm differs from the

previously proposed approaches is several ways. First, a second-order Taylor approximations of the pseudo-outputs were used to obtain an analytical solution. Second, the transformation from the parameter space to the pseudo-parameter space was non-linear and the inverse transformation was obtained numerically. Finally, an explicit equation for computing the sensitivity of the output with respect to the pseudo-parameters at specific sampling times was obtained.

A detailed comparison of the performance of the linear and non-linear algorithms was done by evaluating their WSSE, a measure of output prediction, and computational time, a measure of computational complexity. The non-linear algorithm gives lower WSSE even with a shorter computational time. Finally, the sensitivity of the outputs with respect to the pseudo-parameters was used to develop a D-optimal experimental design which can be used to estimate each of the three pseudo-parameters. The three pseudo-parameters were found to produce a unique local minimum for the WSSE, however, the WSSE was found to have several local minima. This implies that a global optimization procedure should be used in order to compute a global minimum.

References

Audoly, Stefania, Guiseppina Bellu, Leontina D'Angiò, Maria Pia Saccomani, and Claudio Cobelli. "Global identifiability of nonlinear models of biological systems." *IEEE Transactions on Biomedical Engineering* 48, no. 1 (January 2001).

Bard, Yonathan. *Nonlinear Parameter Estimation.* Cambridge, Massachusetts: Academic Press, 1974.

Bastin, G, and D Dochain. *On-line Estimation and Adaptive Control of Bioreactors.* Elsevier Science, 1990.

Ben-Zvi, Amos. "Reparameterization of inestimable systems with applications to chemical and biochemical reactor systems." *AIChE Journal* 54, no. 5 (2008): 1270-1281.

Ben-Zvi, Amos, Kim McAuley, and James McLellan. "Identifiability study of a liquid–liquid phase-transfer catalyzed reaction system." *AIChE Journal* (Wiley InterScience) 50 (2004): 2493–250.

Berland, B R, D J Bonin, O Guerin-Ancey, and N J Antia. "Concentration requirement of glycine as nitrogen source for supporting effective growth of certain marine microplanktonic algae." *Marine Biology* 55, no. 2 (1979): 83-92.

Caperon, J, and J Meyer. "Nitrogen-limited growth of marine phytoplankton– II. Uptake kinetics and their role in nutrient limited growth of phytoplankton." *Deep-Sea Research* 19, no. 9 (1972): 619–632.

Caperon, J, and J Meyer. "Nitrogen-limited growth of marine phytoplankton-I. Changes in population characteristics with steady-state growth rate." *Deep-Sea Research* 19, no. 9 (1972): 601-618.

Chan, Tony F. "An improved algorithm for computing the singular value decomposition." *ACM Transactions on Mathematical Software (TOMS)* (ACM) 8, no. 1 (March 1982): 72-83.

Chen, F, and M Johns. "Effect of C/N ratio and aeration on the fatty acid composition of heterotrophic *Chlorella sorokiniana*." *Journal of Applied Phycology* 3, no. 3 (1991): 203-209.

Chen, G.Q., and F Chen. "Growing phototrophic cells without light." *Biotechnology Letters* 28 (2006): 607-616.

Davidson, K. "Modelling microbial food webs." *Marine Ecology Progress Series* 145 (1996): 279–296.

Davidson, K, and A Cunningham. "Accounting for nutrient processing time in mathematical models of phytoplankton growth." *Limnology and Oceanography* 41, no. 4 (1996): 779–783.

Drapcho, Caye, Nghiem Phu Nhuan, and Terry Walker. *Biofuels Engineering Process Technology.* McGraw-Hill Professional, 2008.

Droop, M R. *Heterotrophy of Carbon.* Vol. 10, in *Algal Physiology and Biochemistry*, by W D P Stewar, 530-559. Blackwell, Oxford.: University of California Press, 1974.

Gotham, I J, and G Y Rhee. "Comparative kinetic studies of nitrate-limited growth and nitrate uptake in phytoplankton in continuous culture." *Journal of Phycology* 17, no. 4 (1981): 309-314.

John, F. *Partial Differential Equations.* New York: Springer-Verlag, 1971.

Kohler, Werner, and Lee Johnson. *Elementary Differential Equations with Boundary Value Problems.* 2nd Edition. Addison Wesley, 2006.

Lee, H Y, L E Erickson, and S S Yang. "Analysis of bioenergetic yield and maintenance parameters associated with mixotrophic and photoheterotrophic growth." *Biotechnology and Bioengineering* 27, no. 12 (1985): 1640-1651.

Li, X, H Xu, and Q Wu. "Large-scale biodiesel production from from microalga *Chlorella protothecoides* through heterotrophic cultivation in bioreactors." *Biotechnology and Bioengineering* 4 (2007): 764-771.

Mailleret, L, J.L. Gouzé, and O Bernard. "Nonlinear control for algae growth models in the chemostat." *Bioprocess and Biosystems Engineering* 27 (2005): 319-327.

Mendes, A, P Guerra, V Madeira, F Ruano, T Lopes da Silva, and A Reis. "Study of docosahexaenoic acid production by the heterotrophic microalga *Crypthecodinium cohnii* CCMP 316 using carob pulp as a promising carbon source." *World Journal of Microbiology and Biotechnology* 29, no. 9 (2007): 1209-1215.

Miao, X, and Q Wu. "Biodiesel production from heterotrophic microalgal oil." *Bioresource Technology* 97 (6) (2006): 841-846.

Minowa, T, S Yokoyama, M Kishimoto, and T Okakura. "Oil production from algal cells of *Dunaliella tertiolecta* by direct thermochemical liquefaction." *Fuel* 74, no. 12 (1995): 1735-1738.

Passarge, J, and J Huisman. "Competition and coexistence." *Ecological Studies* (Springer-Verlag) 161 (2002): 7–42.

Patino, R, M Janssen, and U von Stockar. "A study of the growth for the microalga *Chlorella vulgaris* by photo-bio-calorimetry and other on-line and off-line techniques." *Biotechnology and Bioengineering* 96, no. 4 (2007): 757-767.

Shi, X M, X W Zhang, and F Chen. "Heterotrophic production of biomass and lutein by *Chlorella protothecoides* on various nitrogen sources." *Enzyme and Microbial Technology* 27, no. 3-5 (2000): 312-318.

Sidoli, FR, A Mantalaris, and SP Asprey. "Toward global parametric estimability of a large-scale kinetic single-cell model for mammalian cell cultures." *Industrial & Engineering Chemistry Research* 44, no. 4 (Febuary 2005): 868-878.

Tett, P, and M Droop. *Cell quota models and planktonic primary production.* Vol. 2, in *Handbook of Laboratory Model Systems for Microbial Ecosystems*, by Julian W T Wimpenny, edited by J.W.T. Wimpenny, 177–223. CRC Press, 1988.

USEPA. "A comprehensive analysis of biodiesel impacts on exhaust emissions." United States Environmental Protection Agency., 2002, EPA 420-P-02-001.

Wu, Z Y, C L Shi, and X M Shi. "Modeling of lutein production by heterotrophic *Chlorella* in batch and fed-batch cultures." *World Journal of Microbiology and Biotechnology* 23, no. 9 (2007): 1233-1238.

Yao, ZK, BM Shaw, KB McAuley, and DW Bacon. "Modeling ethelyne/butene copolymerization with multi-site catalysts: Parameter estimability and experimental design." *Polymer Reaction Engineering* 11 (2003): 563-588.

4

Robust Modeling of a Microalgal Heterotrophic Fed-batch Bioreactor.[2]

Abstract

Microalgal feedstock has shown potential for the production of biofuels and fine chemicals. Recently, an optimal experimental input profile for the identification of parameters of a microalgal bioreactor, containing 6 states and 12 unknown parameters has been proposed. In this work, the proposed design is implemented and parameters are estimated. It was found that the parameter estimation procedure can be made more computational efficient by the use of a novel iterative non-linear model reparameterization algorithm. By applying the proposed algorithm to experimental data, a good degree of output prediction was achieved along with bounds on the parameter values. The final, validated, model can be used for optimal control and process simulation.

4.1 Introduction

Recently, there has been increasing worldwide interest in biofuels due to concerns related to climate change and security of energy supply (Peters and Thielmann 2008). Biodiesel, a proven biofuel, provides a cleaner alternative for sustainable energy production when compared to traditional fossil fuels, as it may provide for a 67% reduction of green house

[2] A version of this chapter has been submitted to the Chemical Engineering Science journal.

gases emissions (USEPA 2002) over conventional oil. The use of biodiesel considerably decreases tailpipe particulate matter (-10.1%), unburned hydrocarbon (-21.1%) and carbon monoxide (-11%) emissions when compared to emissions from regular diesel fuel (USEPA 2002). Furthermore, biodiesel is biodegradable and non-toxic. However, land availability is a major limitation for the production of biodiesel from crops such as corn and soybeans which naturally conflicts with food production (Peters and Thielmann 2008). Therefore, limited crop area availability make it infeasible for biodiesel (produced from crops) to meet existing energy demand (Christi 2007).

One viable alternative to crops for the production of biodiesel is microalgae, which are efficient biological systems with a higher biomass production and faster growth rate than other energy crops ((Minowa, et al. 1995), (Miao and Wu 2006)). Moreover, production of microalgae does not require the use of crop area, as it can be grown in desertic areas or intensively in bioreactors. Microalgae have the ability to produce and store high amount of oil that can be used for the production of biodiesel (Li, Xu and Wu 2007) or as single cell oils (Chen and Chen 2006). Oil content in microalgae ranges from 11% to 77% (Christi 2007) with variations due to strain genetics and culture conditions. Currently, biodiesel from algae is not economically competitive with biodiesel produced from conventional sources. The economic competitiveness of algal-based biodiesel production can be improved by increasing the rate at which algae grow and accumulate oil. One approach to improving biodiesel productivity is to implement an optimal model-based control strategy. To successfully implement a model-based control strategy, one must obtain a reliable model of a microalgal bioreactor system. However, microalgal systems are highly non-linear and many of the parameters of bioreactor models cannot be directly obtained from measurement (Audoly, et al. 2001). Therefore, estimating these parameters requires model reparameterization and well designed experiments.

In order to describe bioreactor systems more than 50 different models have been proposed in literature (Bastin and Dochain 1990). However, most control applications have focused on the use of Monod and Haldane models (Wang, Krstic and Bastin 1999). For algal systems, Monod-like models are considered unsuitable because the assumption of growth rate being directly related to extracellular nutrient concentration is not valid for nitrogen and other nutrients (Tett and Droop 1988). Instead, the growth rate depends on the intracellular concentration, or quota (q) (Mailleret, Gouze´ and Bernard 2005). Cell-

quota models, such as Droop (1973) and Caperon-Meyer (1972), have been used to describe the dynamics of algal growth. The Caperon-Meyer model improves on the Droop model by introducing the concept of minimum extracellular nutrient concentration for nutrient uptake (Caperon and Meyer 1972). In the model proposed by Surisetty *et. al.* (Submitted to Chemical Engineering Science), the Caperon-Meyer growth rate kinetic equation was modified in order to account for the simultaneous effect of carbon and nitrogen concentration. Furthermore, the Caperon-Meyer uptake rate expression was modified to reduce the computational requirements while preserving the prediction capability of the kinetic equation. In this work, parameters in the model proposed by Surisetty *et. al.* (Submitted to Chemical Engineering Science) are estimated using experimental data.

In order for a given mathematical model to have any practical applications, the model parameters need to be estimated. Parameter inestimability is a significant obstacle to parameter identification in large non-linear systems and has been well investigated in several fields of research, such as copolymerization with multi-site catalysts (Yao, et al. 2003), ecological systems (Marsili-Libelli, Guerrizio and Checchi 2003), and membrane fuel cells (Corrêa, et al. 2005). Furthermore, in order to gain the maximum amount of information from an experiment, one can implement an optimal experimental design. This results in the number of estimable parameters being maximized. Methods for the optimal experimental design for biological systems have been proposed by Wu (2008) and Balsa-Canto (2008).

In a previous work, a first principles-based microalgal bioreactor ODE model, containing 6 states and 12 unknown parameters was been proposed (Surisetty, et al. Submitted to Chemical Engineering Science). A sensitivity-based approach was used in order to identify parameters which have a minimal effect on model behaviour. No single parameter was seen to have negligible effect on output prediction; however, directions in the parameter space along which there is minimal change in the output predictions were observed. Furthermore, a non-linear transformation based model reparameterization algorithm was proposed in order to reduce the dimension of the parameter space to facilitate the use of a first principles ODE model for online control and optimization (Surisetty, et al. Submitted to Chemical Engineering Science). Implementing the non-linear algorithm, via simulations, showed that it produced good output prediction

capability at a drastically lower computational cost. This was achieved by reducing the number of estimated parameters from twelve original parameters to the three pseudo-parameters. Finally, a D-optimal experimental input profile for the identification of the three pseudo-parameters was proposed.

In this work, the experiment design proposed by Surisetty *et. al.* (Submitted to Chemical Engineering Science) was implemented and used to estimate parameters in the microalgal bioreactor model. Experimental data on total biomass concentration, external nitrogen source concentration, external carbon source concentration, and total algal oil stored in cells was collected and analyzed. In order to achieve a good degree of output prediction, the algorithm had to be modified by introducing an iterative step. The iterative non-linear algorithm was able to greatly decrease computational time while achieving a good degree of matching between model predictions and experimental data.

4.2 Microalgal bioreactor model

A first principles ODE model of an algal bioreactor system with an intracellular product is considered in this work (Surisetty, et al. Submitted to Chemical Engineering Science). This model contains three system inputs, 12 unknown parameters and the following 6 system states:

7) x: Functionally active biomass concentration [g/mL]
8) S_1: Nitrogen source concentration in culture media [g/mL]
9) S_2: Carbon source concentration in culture media [g/mL]
10) Q: Total nitrogen cellular quota [g/mL]
11) I_p: Total algal oil stored in cells [g/mL]
12) V: Total reaction volume [mL]

The dynamics of the bioreactor are described by a set of 6 differential equations shown in Table 4.1. The measured quantities are external carbon source concentration (S_2), external nitrogen source concentration (S_1), measurable biomass concentration (X), and mass fraction of oil stored in cells $\left(\frac{I_p}{X}\right)$. The measurable biomass concentration (X) is defined as follows:

$$X = x + I_p + Q \tag{4.1}$$

The three system inputs are flowrates of the nitrogen and carbon rich feeds (denoted by f_1^i and f_2^i respectively) and the outflow from the reactor, f_o. For the fed-batch reactor system considered in this work, f_o profile was not used in the experimental design; rather, it is used to account for the change in reactor volume due to sampling. The governing system rates are biomass growth rate (μ), nitrogen uptake rate (ρ), and oil production rate (π). The formulations of μ, ρ and π are given in Equation 4.2, where $q = Q/X$. The definitions of the parameters in Equation 4.2 are given in Table 4.2. The initial guesses for parameter values used in the output prediction algorithms are also given in Table 4.2. A detailed derivation of this model is available in Surisetty *et. al.* (Submitted to Chemical Engineering Science).

$$\mu = \mu_m \left(\frac{q - q_m}{K_q + q}\right)\left(\frac{S_2}{K_s + S_2}\right)$$

$$\rho = \rho_m \left(1 - \frac{S_0}{S_1}\right)^{1+\varepsilon} \tag{4.2}$$

$$\pi = \pi_m \frac{S_2}{K_\pi + S_2}\left(1 - \frac{I_p}{X}\right)$$

Table 4.1. Continuous bioreactor model

$\frac{dx}{dt} = \mu x - x\frac{f_o}{V} - x(f_1^i + f_2^i - f_o)\frac{1}{V}$	(f_1)
$\frac{dS_1}{dt} = -\rho x + S_1^i\frac{f_1^i}{V} - S_1\frac{f_o}{V} - S_1(f_1^i + f_2^i - f_o)\frac{1}{V}$	(f_2)
$\frac{dS_2}{dt} = -\frac{1}{Y_{xs}}\mu x + S_2^i\frac{f_2^i}{V} - S_2\frac{f_o}{V} - k_m x - \frac{1}{Y_{ps}}\pi x - S_2(f_1^i + f_2^i - f_o)\frac{1}{V}$	(f_3)
$\frac{dQ}{dt} = \rho x - \frac{1}{Y_{xq}}\mu x - Q\frac{f_o}{V} - Q(f_1^i + f_2^i - f_o)\frac{1}{V}$	(f_4)
$\frac{dI_p}{dt} = \pi x - I_p\frac{f_o}{V} - I_p(f_1^i + f_2^i - f_o)\frac{1}{V}$	(f_5)
$\frac{dV}{dt} = f_1^i + f_2^i - f_o$	(f_6)
System outputs	
$y(t) = \begin{bmatrix} X & S_1 & S_2 & \frac{I_p}{X} \end{bmatrix}^T$	(y)

Table 4.2. Definitions and initial guesses for parameters

Number	Parameter	Name	Initial guess	Lower Limit	Upper Limit	Unit
P1	μ_m	Maximum growth rate	0.15	0.005 (2)	0.46 (2)	1 / h
P2	q_m	Minimum cell quota for supporting growth	0.027	0 (3)	0.064 (2)	g / g
P3	K_q	Half saturation constant of nitrogen quota for growth	0.5	0.001 (3)	0.6 (3)	g / g
P4	ρ_m	Maximum uptake rate	0.08	0.00045 (2)	0.204 (2)	1 / h
P5	K_s	Half saturation constant of carbon source for growth	0.014	0.001 (3)	0.12 (2)	g / mL
P6	s_o	Threshold substrate concentration	0	0 (1)	0.0007 (2)	g / mL
P7	$1/Y_{xs}$	Inverse of Biomass to substrate yield	1.8	1.5 (1)	16.7 (2)	g / g
P8	k_m	Maintenance constant	0	0 (2)	0.02 (3)	g / g h
P9	$1/Y_{ps}$	Inverse of product to substrate yield	2.5	1.9 (1)	10 (3)	g / g
P10	π_m	Maximum oil production rate	0.05	0.001 (3)	0.1 (3)	1 / h
P11	K_π	Half saturation constant for oil production	0.01	0.001 (3)	0.1 (3)	g / mL
P12	$1/Y_{xq}$	Inverse of biomass to substrate quota yield	0.03	0.0138 (2)	0.282 (2)	g / g

(1). Case 1. Picked based on stoichiometric relationships.

(2). Case 2. Limits are chosen based in 50% of the literature minimum or 200% of the literature maximum.

(3). Case 3. Initially picked based on Case 1; however, bounds were relaxed as they were hit during optimization.

In order to implement the parameter estimation algorithm, it is necessary to define the range of feasible parameter values. The feasible ranges for parameter estimates can be restricted by examining the physical interpretation of the parameters. The lower and upper limits used for estimation are given in Table 4.2. These limits were chosen using three different criteria (labeled Case 1, 2, and 3). In all cases, parameters must be non-negative; parameters representing molar or mass fraction (q_m and K_q) are bounded from 0 to 1. In Case 1, the yields (Y_{xs} and Y_{ps}) are constrained by stoichiometric relationships. Such physical restrictions introduce hard limits in the upper and lower bounds of the parameter values. To further reduce the feasible parameter set, soft constraints were introduced (Case 2). For example, parameters representing maximum reaction rates or saturation constants do not have a theoretical maximum value, but usually their feasible maximum values are known literature (Surisetty, et al. Submitted to Chemical Engineering Science). In Case 2, the bounds were conservatively picked to be half the minimum (lower bound) and twice the maximum (upper bound) values reported in literature. To ensure that the parameter estimates remained in the interior of the feasible region, the limits obtained from Case 2 were loosened if a soft constraint was violated. In such a situation, the new parameter limits were relaxed sufficiently so that none of the soft limits were violated (the new relaxed limits are labelled Case 3 in Table 4.2).

The stoichiometric relationship used to constraint the parameters in Case 1 were obtained as follows. For $1/Y_{ps}$, it is assumed that all the oil in the cells is tri-palmitoyl-glycerol ($C_{51}H_{98}O_6$), which represents a high percentage of the oil in algal cells. Assuming a 100% conversion of the carbon present in glucose ($C_6H_{12}O_6$) to oil we have a maximum mass yield equal to 0.526, inverse of which is 1.9. In the case of $1/ Y_{xs}$, consider that the utilization of glucose, as the energy source of the cell, requires the conversion of glucose to pyruvate and further processing to acetyl-CoA. The free-energy released in these transformations is stored as ATP and NADH, which will further be used as the energy carriers for all the metabolic processes in the cell. In the conversion from glucose to acetyl-CoA, two molecules of CO_2 are produced per molecule of glucose being converted, therefore the maximum carbon yield in molar basis is 0.667. Here a minimum carbon content in the cells equals to 0.4 (w/w, dry basis) is considered, as typical carbon content in microalgae ranges from 0.47 to 0.61 (w/w, dry basis) (El-Sarraf and El-

Shaarawy 1994). Therefore, the maximum glucose to biomass yield in mass basis is 0.667.

4.3 Materials and methods

4.3.1 Organism and medium

The fresh-water green microalgae *Auxenochlorella protothecoides*, UTEX 25 (UTEX Culture Collection of Algae, Texas), were cultivated axenically under heterotrophic conditions. *A. protothecoides* cultures were maintained on agar plates with agar medium ATCC 5. Cells from agar plates were suspended in 200 mL seed medium, and incubated at 25°C and 100 rpm. After 96 h of incubation, 10 mL of algal suspension was transfer to inoculation medium, and incubation procedure repeated.

The seed and inoculation media contained: KH_2PO_4 (2.8 g/L), K_2HPO_4 (1.2 g/L), $MgSO_4{\cdot}7H_2O$ (1.2 g/L), $FeSO_4{\cdot}7H_2O$ (48 mg/L), H_3BO_3 (11.6 mg/L), $CaCl_2{\cdot}2H_2O$ (10 mg/L), $MnCl_2{\cdot}4H_2O$ (7.2 mg/L), $ZnSO_4{\cdot}7H_2O$ (0.88 mg/L), $CuSO_4{\cdot}5H_2O$ (0.32 mg/L), MoO_3 (72 μg/L), thiamine hydrochloride (40 μg/L), glucose (40 g/L, Sigma Aldrich), and glycine (0.5 g/L). All reagents were of analytical grade obtained from Fisher Scientific Co, except where noted. Solutions were prepared using deionized water (Milli-Q, Millipore).

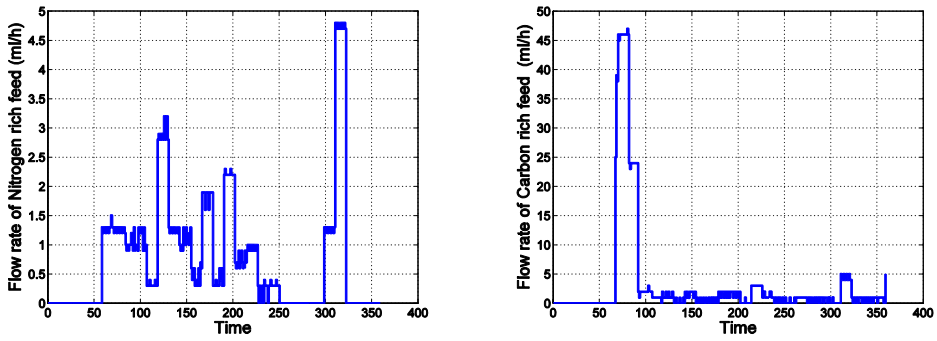
4.3.2 Bioreactor equipment and conditions

Experiments were conducted in a 3L stirred-tank bioreactor (Sartorious Biostat A plus, working volume 2L), using 10% inoculum. Start-up medium had the same composition as the inoculation medium. For the fed-batch operation, two feeds were added at predefined times as calculated from the D-optimal algorithm. Feed 1, f_1^i, contained only glycine, which was the sole nitrogen source, at a concentration of 10 g/L. Feed 2, f_2^i, contained glucose and the same minerals present in the start-up medium, except KH_2PO_4 and K_2HPO_4, at a concentration equal to five times that of the start-up medium.

Bioreactor temperature was kept at 25°C using an electrical heating blanket; pH was controlled at 6.2 by supplementing KH_2PO_4 (acid) and K_2HPO_4 (base) as required; and

agitation rate was kept constant at 300 rpm, while varying aeration rate to control dissolved oxygen at around 30% of the saturation value.

Feed streams were pumped into the bioreactor using a set of peristaltic pumps (Sartorious BBI and 101 U/R Watson Marlow). Feed flow-rates were varied every hour according to an optimal input signal as described by Surisetty *et. al.* (Submitted to Chemical Engineering Science). Flowrate set points were sent from MATLAB® to the bioreactor control unit (MFCS/win, Sartorius BBI) using an in-House OPC client (OLE for Process Control). Figure 4.1 shows the implemented input profiles.



(a) (b)

Figure 4.1. Input profiles. (a) Flow rate of nitrogen rich feed (b) Flow rate of carbon rich feed

An automatic sampler system was set-up in order to withdraw 5.5mL of culture broth every four hours, and store it in a fridge at 4°C for further analysis. A process schematic of the experimental setup is given in Figure 4.2.

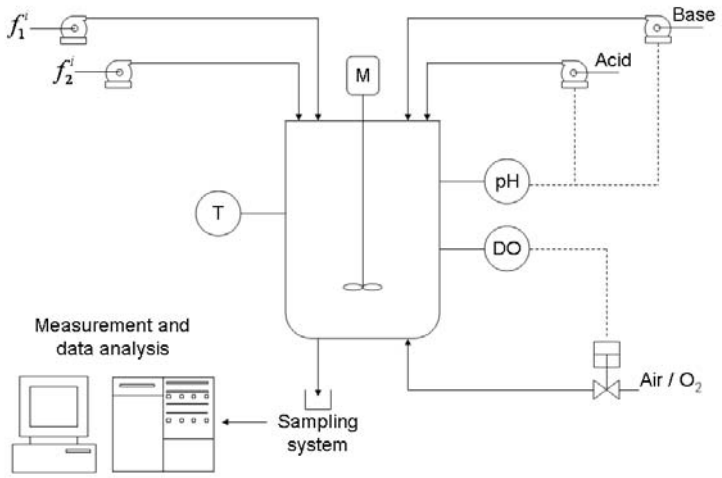


Figure 4.2. Process schematic of the experimental setup.

4.3.3 Analytical methods

Biomass concentration was determined as total suspended solids (TSS) by centrifuging 1.4 mL of cell suspension at 10000 rpm for 10 minutes. Pellets were washed twice with a saline phosphate buffer solution (pH 6.2) and recentrifuged. Final precipitate was vacuum dried at 50°C and 0.1 bar until constant weight. These measurements were checked against vacuum filtration of 10 mL of the culture broth with a 0.22 μm filter paper (Whatman #5). The average difference among duplicates in dry biomass measurements was below 2.4%.

The clear supernatant from the centrifugation was filtered using a 0.22 μm syringe filter in order to remove any residual cells. Glucose concentration in this filtered supernatant was measured by high performance liquid chromatography (Agilent 1200 Series HPLC), with a SupelcoGel Pb carbohydrate column at 70°C (Internal diameter 7.8mm, length 30cm) and a guard column. Sample injection volume was 10 μL; eluent was deionized, sterile water (Milli-Q, Millipore); elution flow-rate was set at 0.5 mL/min, and a refractive index detector (RID) at 35°C was used. Glucose solution standards with concentration ranging from 0.1 g/L to 100 g/L were prepared and analyzed by HPLC, based on which a calibration curve was defined. The average error in HPLC measurements was 0.5%.

Total nitrogen in the filtered supernatant was determined by pyro-chemiluminescence using an Antek 9000NS nitrogen analyzer. Sample injection volume was 20μL; furnace was set up at 1050°C; gas flowrates were helium 140 mL/min, and oxygen 450 mL/min. Glycine was used as standard for calibration. The relative standard deviation of the measurements was 2%.

Oil content in the cells was determined by flurospectrometry of cells stained with Nile Red (Chen, et al. 2009). In this method, fluorescence intensity is linearly correlated to the total neutral lipid content of the cells. A 10μL aliquot of a 10μg/L Nile Red solution in ethanol was added to the individual wells of a 96-microplate containing 10μL samples of 10 g/L algal cells. Volume in each well was completed to 200 μL by adding a 30% (v/v) ethanol solution in water. Samples were incubated at 40°C for 10 min, and fluorescence emissions were recorded with a multiplate reader spectrophotometer (Fluoroskan Ascent). Excitation and emission wavelength were selected at 536 nm and 604 nm, respectively.

4.4 Parameter estimation

4.4.1 Model reparameterization

The non-linear algorithm for model reparameterization for the purpose of output prediction is based on a transformation approach proposed by Ben-Zvi (2008) for reparameterizing unidentifiable or inestimable systems. In the non-linear algorithm, the parameter space is partitioned into estimable and inestimable spaces using a non-linear transformation, which utilizes differential geometry and *a priori* knowledge of the process quantities having a significant effect on model behavior. In the work done by Surisetty *et. al.* (Submitted to Chemical Engineering Science), it was shown that choosing the set of pseudo-parameters (denoted by $\phi^{+}(P)$) as non-linear functions of the three system rate functions (μ, ρ and π) provides for accurate model predictions at a reduced computational cost.

Under the parameter estimation framework proposed by Surisetty *et. al.* (Submitted to Chemical Engineering Science), $\phi^{+}(P)$ represents the estimable part of the parameter space. After choosing $\phi^{+}(P)$, a set of coordinates $\phi^{-}(P)$ must be computed such that

they form a basis for the inestimable part of the parameter space. Calculating $\phi^{-}(P)$ quickly increases in complexity with more parameters, as it requires solving a set of linear partial differential equations of size $d * d_1$, where d and d_1 are the total number of system parameters and pseudo-outputs respectively. In the case where the method of characteristics (John 1971) is used to solve the set of linear partial differential equations, a set of non-linear ODEs need to be solved analytically, which could not be obtained using MAPLE™ 11.0. Therefore, a second-order Taylor series approximation, about certain parameter values (point of approximation), of $\phi^{+}(P)$ is used to calculate $\phi^{-}(P)$. The end result is a non-linear transformation, *M*, which transforms the actual parameter space, *P*, to a pseudo-parameter space, τ, is $P = M(\tau)$. A detailed description of this approach is available in Surisetty *et. al.* (Submitted to Chemical Engineering Science). Note that the values of τ used in this approach had to be constrained so that the resulting parameters $P = M(\tau)$ were within the bounds given in Table 4.4.

In order to increase the rate at which the algorithm converges, a modification was introduced. Under this modification, the second order Taylor series approximation for $\phi^{+}(P)$ was recomputed as improved parameter estimates became available. Figure 4.3 contains a flowchart describing the procedure of the iterative non-linear algorithm. The optimization problem in each iteration step is formulated as follows:

$$[\tau_1 \dots \tau_3] = \underset{\tau_1 \dots \tau_3}{\operatorname{argmin}} \sum_{t=1}^{t_f} (y_t - \tilde{y}_t)^2 \tag{4.3}$$

where $\tilde{y}$ is the predicted output based on the pseudo-parameters $\tau_1 = \phi_1^{+}(P), \tau_2 = \phi_2^{+}(P)$ and $\tau_3 = \phi_3^{+}(P)$, y_t is the experimental output and t_f is the final time. The iteration loop is stopped when the difference between the current iteration Weighted Sum of Squared Error (WSSE), and the previous iteration WSSE is less than a pre-specified threshold (a threshold value of 0.05 was used in this work).

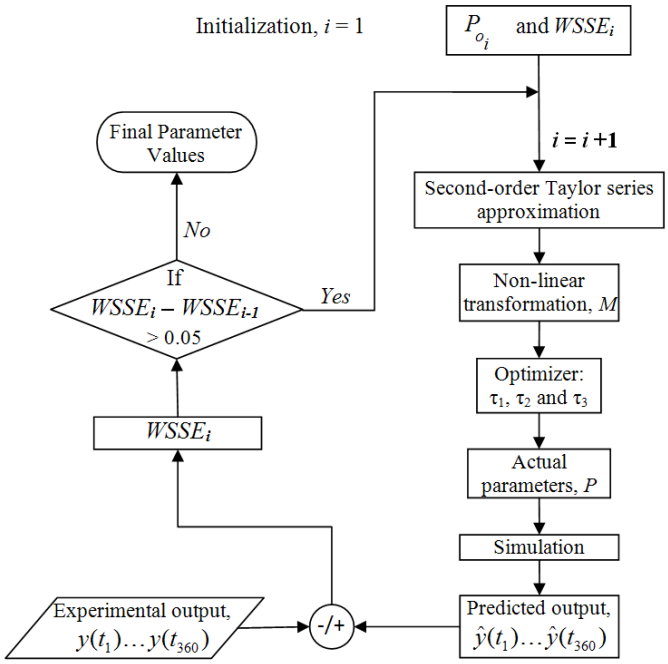


Figure 4.3. A flowchart of the iterative non-linear algorithm.

In order to equivalently weigh the error in different outputs, the diagonal entries of the weighing matrix, W, were chosen to be proportional to the inverse of the average value of the output in the WSSE calculation.

$$W = \begin{bmatrix} 55.71 & 0 & 0 & 0 \\ 0 & 9037.28 & 0 & 0 \\ 0 & 0 & 20.01 & 0 \\ 0 & 0 & 0 & 2.27 \end{bmatrix} \tag{4.4}$$

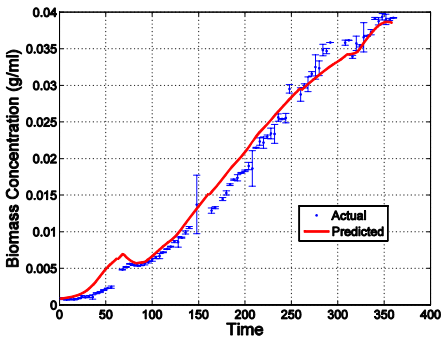
The results of output prediction from the iterative non-linear algorithm are shown in Figure 4.4 and Table 4.3. As shown in Figure 4.4, the predictions obtained by using this algorithm are able to capture the trends exhibited by the experimental bioreactor system. The computational time of the iterative non-linear algorithm was 1 hour and 29 minutes on a system with a AMD Athlon(tm) 64 FX-51 Processor with 2.2 GHz with 2 GB RAM.

In order to validate the proposed reparameterization approach, parameter estimates for the full twelve parameter model were obtained and compared to the model predications and parameter estimates using the iterative non-linear reparameterization algorithm. Using the parameter values given in Table 4.2 as the initial guess, for estimating all twelve parameters, it was not possible to obtain accurate model predictions in less than

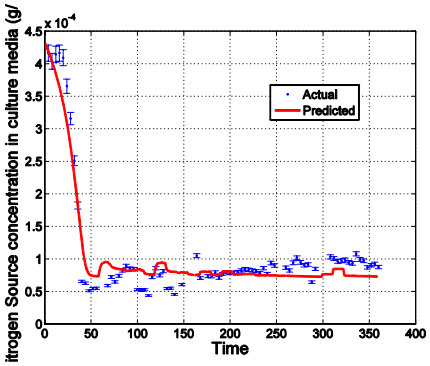
60 hours of computation time. While the twelve parameter model could not be used to obtain a global optimum, it is possible that this model could be used to improve, locally, on the estimates obtained from the iterative non-linear algorithm. To test whether this is the case, the parameter estimates from the iterative non-linear algorithm were used as an initial guess for the identification of the full model. Optimizing over all twelve parameters locally improved the WSSE from 5.1990 to 5.1840, a difference of 0.3%. The WSSE and the computational time of the two scenarios are given in Table 4.3. Note that the improvement in WSSE was less that the stopping threshold using in the iterative algorithm. In addition optimizing over all twelve parameters required an additional computational time of 49 hours and 12 minutes as compared to the 1 hour and 29 minutes of computational time for the iterative non-linear algorithm. Therefore, the slight improvement in WSSE obtained by estimating all twelve parameters is unlikely to be offset by the additional computational burden.

Table 4.3. Weighted sum of squared error, WSSE, of proposed algorithms

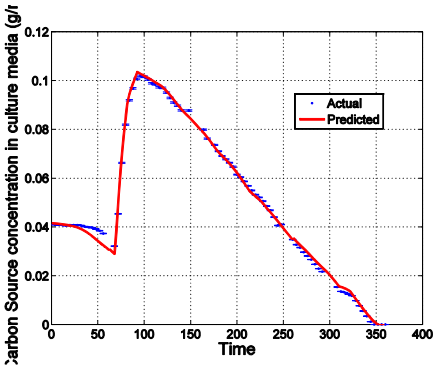
Scenario	Description	WSSE	Computational time
1	Estimating all 12 parameters	5.1840	49 hours and 12 minutes
2	Iterative non-linear algorithm	5.1990	1 hour and 29 minutes



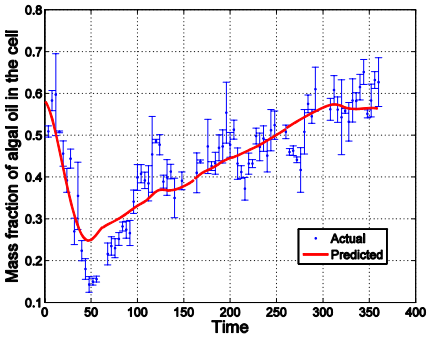
(a)



(b)



(c)



(d)

Figure 4.4. Comparison of experimental and predicted outputs (iterative non-linear algorithm). (a) $y_1 = X$ (b) $y_2 = S_1$ (c) $y_3 = S_2$ (d) $y_4 = \frac{I_p}{X}$

The parameter estimates were achieved by varying only the key parameters $\tau_1 \ldots \tau_3$ and keeping $\tau_4 \ldots \tau_{12}$ constant. In order to investigate whether any of $\tau_4 \ldots \tau_{12}$ do have a strong effect on model predictions, each parameter was varied, in turn, over the range $[0 \quad 2\tau_i]$ for $i \in \{1,2,\ldots,12\}$. The minimum WSSE for each τ_i is given in Table 4.4 along with the percent change between the estimated and the optimal value of τ_i (denoted by $\Delta\tau$ in Table 4.4). Note that, by definition, $\Delta\tau_1 = \Delta\tau_2 = \Delta\tau_3 = 0$.

Table 4.4. Effect of each τ_i on WSSE

τ	$\Delta\tau$ (%)	Minimum WSSE
1	0	5.199
2	0	5.199
3	0	5.199
4	0.2	5.199
5	0.1	5.199
6	0	5.199
7	-17.1	5.1894
8	1.4	5.195
9	1.9	5.1971
10	0.2	5.199
11	-5.2	5.1861
12	0.1	5.199

It can be seen from Table 4.4 that the minimum WSSE achievable by varying values of $\tau_4 \ldots \tau_{12}$ is very close to the WSSE of the iterative non-linear algorithm (Table 4.3). As a result, the improvement in the WSSE may not be significant enough to offset the increase

in computational cost resulting from estimating additional pseudo-parameters. These results reinforces the conclusion, obtained by attempting to estimate all 12 model parameters, that attempting to estimate additional parameters will not significantly improve model predictions.

4.4.2 Model parameters

In the iterative non-linear algorithm, the pseudo-parameters are chosen as non-linear functions of the three system rate functions. As a result, using this algorithm, one can only be sure of the estimated rate functions and not the values of the specific model parameters. The rate functions depend on both system states and parameters, as a result, the values of the rate functions given in Table 4.5 are based on estimated parameter values and the nominal state conditions given in Table 4.5. These nominal state values are evaluated using the average observed output values. The rate functions at different state values can be calculated by using Equation 4.2.

Table 4.5. Estimated values of the kinetic rate functions at the nominal state values

Rate	Description	Value	Unit
μ	Biomass growth rate	0.0065	1 / h
ρ	Nitrogen uptake rate	0.0040	1 / h
π	Oil production rate	0.0088	1 / h
State	**Description**	**Nominal value**	**Unit**
x	Functionally active biomass concentration	0.00457	g/mL
S_1	Nitrogen source concentration in culture media	1.106e-4	g/mL
S_2	Carbon source concentration in culture media	0.050	g/mL
Q	Total nitrogen cellular quota	0.00546	g/mL
I_p	Total algal oil stored in cells	0.00792	g/mL
V	Total reaction volume	1790	mL

In Table 4.6, the estimated parameter values are given. In order to assess the uncertainty associated with these estimates, a perturbation-based test was used. Note that approximate 95% confidence intervals for the system parameters based on a Gaussian distribution (e.g. a *t*-test to determine 95% confidence intervals) could not be used

because of the non-linearity of the model equations. For example, the 95% confidence interval for parameter 1 was (-1.897, 2.538) which is not correct because parameter 1 must be positive.

A simple approach to assess the degree of uncertainty in each parameter is to obtain a change in each parameter that is necessary to cause a 10% deviation in the WSSE. The range for each parameter value, using this approach, is given in Table 4.6. Also given in Table 4.6 is the percent change from estimated parameter value at the high and low bounds of the uncertainty region for each parameter. Note that, a value of zero was used constraint the lower bound of the parameter uncertainty region. Also, if doubling of a parameter value did not result in a 10% change in the WSSE, the upper bound for uncertainty region was not estimated. In this case, the parameter was deemed inestimable using the available data set.

Table 4.6. Estimated actual parameter values

Number	Parameter	Estimated value	Range			
			Lower		Upper	
			Value	% change	Value	% change
P1	μ_m	0.3204	0.2675	-16.5	0.405	26.4
P2	q_m	1.672e-4	0	-100	-	-
P3	K_q	0.2450	0.1915	-23.4	0.307	22.8
P4	ρ_m	0.01219	0.011	-9.6	0.0136	11.4
P5	K_s	0.0010	0	-100	-	-
P6	s_o	7.402e-5	6.5365e-5	-11.7	8.179e-5	10.5
P7	$1/Y_{xs}$	1.5002	1.1447	-23.7	1.9278	28.5
P8	k_m	0.01385	0.0112	-19.5	0.017	22.5
P9	$1/Y_{ps}$	2.2802	1.9656	-13.8	2.6702	17.1
P10	π_m	0.01640	0.0146	-11.1	0.0193	17.4
P11	K_π	0.001792	0	-100	-	-
P12	$1/Y_{xq}$	0.07370	0.0688	-6.6	0.0772	4.8

4.5 Improvements in optimization

To quantify the possible benefits of modeling the microalgal bioreactor, the oil yield in the following two scenarios was compared. In Scenario 1, the parameter values that are given in Table 4.2 (initial guess) were used to compute the input profile which maximizes volumetric oil productivity. In Scenario 2, estimated parameters in Table 4.6 were used to compute the optimal input profile. The optimized input profiles from both scenarios were

simulated on the same system based on the estimated parameter values listed in Table 4.6. Figure 4.5 shows that the simulated oil concentration for Scenario 2 is 51% higher than that of first scenario after 150 hours. Therefore, if one assumes that the estimated parameters in Table 4.6 explain the dynamics of the microalgal bioreactor system, then effort taken to identify these parameters would yield a 51% increase in volumetric productivity.

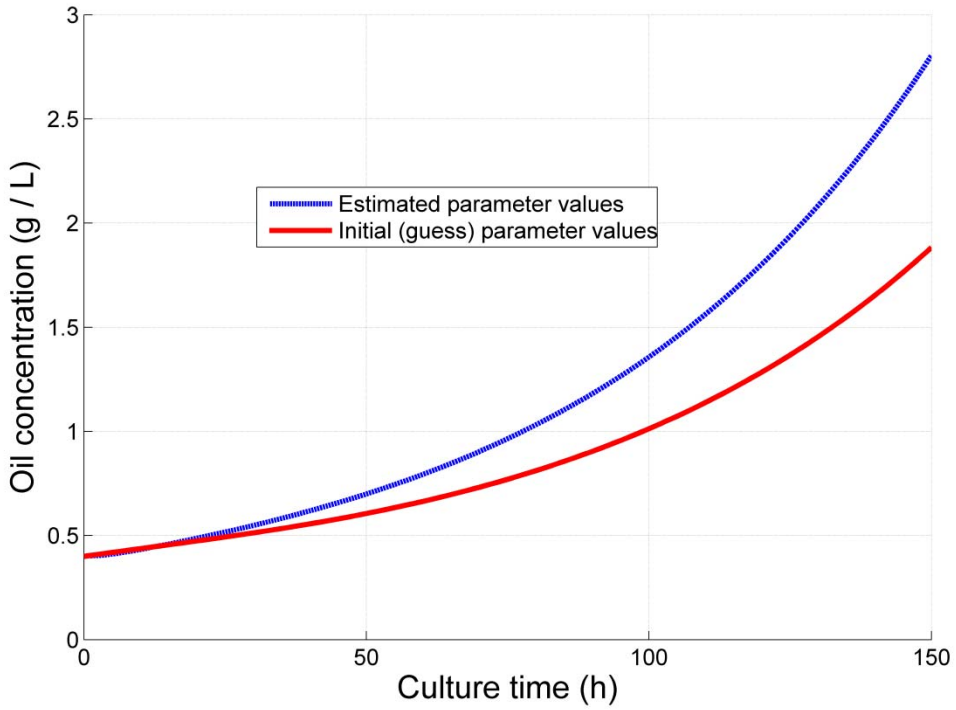


Figure 4.5. Simulated oil profiles based on optimal inputs calculated using estimated and initial (guess) parameter values.

4.6 Conclusions

Biodiesel is an alternative to traditional fossil fuels. Production of biodiesel from crops is greatly limited by acreage availability. Microalgae can be grown, in high concentrations, intensively in bioreactors. As a reuslt, microalgae can be grown without displacing crop land. Furthermore, when compared to other feedstock for biodiesel, microalgae have higher biomass production and faster growth rate (Minowa, et al. 1995).

Previously, a microalgal bioreactor model was proposed (Surisetty, et al. Submitted to Chemical Engineering Science). This first principles-based ODE model contains 6 states and 12 unknown parameters. Using a sensitivity-based approach, it was determined that there are directions in the parameter space along which minimal change in the output predictions were observed. Using this insight, a non-linear transformation based model reparameterization algorithm was proposed in order to reduce the number of estimated variables from twelve parameters to three pseudo-parameters. This algorithm was shown to significantly decreasing computational time while not degrading the prediction capability of the model. A feed profile which is D-optimal with respect to the three pseudo-parameters from the non-linear algorithm was proposed.

In this work, the D-optimal feed profile computed by Surisetty *et. al.* (Submitted to Chemical Engineering Science) was implemented in a lab-scale microalgal bioreactor system. Experimental data for external carbon source concentration, external nitrogen source concentration, measurable biomass concentration, and mass fraction of oil stored in cells were collected every four hours and analyzed.

In order to achieve satisfactory model predictions, the algorithm was modified such that at each iteration the point of approximation is updated as improved parameter estimates become available. This iterative non-linear algorithm was able to achieve an adequate degree of output prediction; a set of estimated parameter values is reported.

In order to validate the proposed reparameterization approach, the WSSE of the iterative non-linear algorithms was compared to the WSSE obtained by estimating all twelve parameters. Accurate model predictions could not obtained when the parameter values given in Table 4.2 were used as an initial guess for estimating all twelve parameters. Furthermore, even when the parameter estimates obtained using the iterative algorithm were used as an initial guess for estimating all twelve parameters, the improvement in WSSE was negligible. Furthermore, a perturbation-based test was used to show that reducing the number of estimated variable from twelve to three did not significantly degrade model predictions. The possible benefits of obtained an accurate model for algal growth are quantified in terms of increased volumetric productivity of oil under optimal control.

References

Audoly, Stefania, Guiseppina Bellu, Leontina D'Angiò, Maria Pia Saccomani, and Claudio Cobelli. "Global identifiability of nonlinear models of biological systems." *IEEE Transactions on Biomedical Engineering* 48, no. 1 (January 2001).

Balsa-Canto, E, A.A. Alonso, and J.R. Banga. "Computational procedures for optimal experimental design in biological systems." *IET Systems Biology* 2, no. 4 (2008): 163-172.

Bastin, G, and D Dochain. *On-line Estimation and Adaptive Control of Bioreactors.* Elsevier Science, 1990.

Ben-Zvi, Amos. "Reparameterization of inestimable systems with applications to chemical and biochemical reactor systems." *AIChE Journal* 54, no. 5 (2008): 1270-1281.

Caperon, J, and J Meyer. "Nitrogen-limited growth of marine phytoplankton– II. Uptake kinetics and their role in nutrient limited growth of phytoplankton." *Deep-Sea Research* 19, no. 9 (1972): 619–632.

Chen, G.Q., and F Chen. "Growing phototrophic cells without light." *Biotechnology Letters* 28 (2006): 607-616.

Chen, Wei, Zhang Chengwu, Lirong Song, Milton Sommerfeld, and Qiang Hu. "A high throughput Nile red method for quantitative measurement of neutral lipids in microalgae." *Journal of Microbiological Methods* 77 (2009): 41-47.

Christi, Y. "Biodiesel from microalga." *Biotechnology Advances* 25 (2007): 294-306.

Corrêa, , Jeferson M, Felix A Farret, Vladimir A Popov, and Marcelo G Simões. "Sensitivity analysis of the modeling parameters used in simulation of proton exchange membrane fuel cells." *IEEE Transactions on Energy Conversion* 20 (2005): 211-218.

Droop, M. R. "Some thoughts on nutrient limitation in algae." *Journal of Phycology* 9 (1973): 264-272.

El-Sarraf, W.M., and G El-Shaarawy. "Chemical composition of some marine algae from the mediterranean sea of Alexandria, Egypt." *The Bulletin of the High Institute of Public Health* 24, no. 3 (1994): 523-534.

John, F. *Partial Differential Equations.* New York: Springer-Verlag, 1971.

Li, X, H Xu, and Q Wu. "Large-scale biodiesel production from from microalga *Chlorella protothecoides* through heterotrophic cultivation in bioreactors." *Biotechnology and Bioengineering* 4 (2007): 764-771.

Mailleret, L, J.L. Gouzé, and O Bernard. "Nonlinear control for algae growth models in the chemostat." *Bioprocess and Biosystems Engineering* 27 (2005): 319-327.

Marsili-Libelli, S, S Guerrizio, and N Checchi. "Confidence regions of estimated parameters for ecological systems." *Ecological Modelling* 165 (2003): 127-146.

Miao, X, and Q Wu. "Biodiesel production from heterotrophic microalgal oil." *Bioresource Technology* 97 (6) (2006): 841-846.

Minowa, T, S Yokoyama, M Kishimoto, and T Okakura. "Oil production from algal cells of *Dunaliella tertiolecta* by direct thermochemical liquefaction." *Fuel* 74, no. 12 (1995): 1735-1738.

Peters, Jörg, and Sascha Thielmann. "Promoting biofuels: implications for developing countries." *Energy Policy* 36 (2008): 1538–1544.

Surisetty, Kartik, Hector De la Hoz Siegler, William C McCaffrey, and Amos Ben-Zvi. "Model re-parameterization and output prediction for a bioreactor system." *Chemical Engineering Science*, Submitted to Chemical Engineering Science.

Tett, P, and M Droop. *Cell quota models and planktonic primary production.* Vol. 2, in *Handbook of Laboratory Model Systems for Microbial Ecosystems*, by Julian W T Wimpenny, edited by J.W.T. Wimpenny, 177–223. CRC Press, 1988.

USEPA. "A comprehensive analysis of biodiesel impacts on exhaust emissions." United States Environmental Protection Agency., 2002, EPA 420-P-02-001.

Wang, Hsin-Hsiung, Miroslav Krstic, and Georges Bastin. "Optimizing bireactors by extremum seeking." *International Journal of Adaptive Control and Signal Processing* 13 (1999): 651-669.

Wu, Fang-Xiang, Lei Mu, and Ruizhi Luo. "Complexity analysis and optimal experimental design for parameter estimation of biological systems." *IEEE Canadian Conference on Electrical and Computer Engineering.* Institute of Electrical and Electronics Engineers Inc., 2008. 393-397.

Yao, ZK, BM Shaw, KB McAuley, and DW Bacon. "Modeling ethelyne/butene copolymerization with multi-site catalysts: Parameter estimability and experimental design." *Polymer Reaction Engineering* 11 (2003): 563-588.

5

Conclusions and Future work

5.1 Conclusions

Biodiesel is a viable alternative to traditional fossil fuels due to characteristics such as non-toxicity and biodegradability. Microalgae have great potential for the production of biodiesel as a result of their higher biomass production at a faster growth rate when compared to traditional feedstocks such as corn and soybean (Minowa, Yokoyama, et al., Oil Production from Algal Cells of Dunaliella tertiolecta by Direct Thermochemical Liquefaction 1995). Furthermore, production of biodiesel from microalgae is not limited by acreage availability as microalgae are highly efficient biological organisms that can be grown intensively in bioreactors.

An ODE microalgal bioreactor model containing six states, three inputs and twelve unknown parameters is proposed. The model was developed based in first-principle conservation equations comprising of three rate functions: biomass growth rate, uptake rate and oil production rate. The model considers biomass, two substrates, intracellular nitrogen concentrations, as well as intracellular product. A sensitivity-based test was used to check for parameter estimability. The parameters were perturbed pair-wise and the resulting output was compared to a nominal output response. The simulations in the sensitivity-based test were based on an input profile that was designed based on the D-optimality criterion. Not all parameters were found to have unique contributions to model behaviour, i.e. some parameters are correlated.

An algorithm based on singular value decomposition of the parameter covariance matrix is proposed to identify directions on the parameter space along which minimal change in model output occurs. A set of eight linear combinations of parameters was recognized.

Using this information, the parameter space was partitioned into estimable and inestimable linear subspaces, as a result of which the number of quantities required to be estimated decreased from twelve to four. Based on initial simulations, it can be seen that the linear model reparameterization algorithm greatly decreased the computational expense while achieving a good degree of output prediction.

From the work done be Ben-Zvi (2008) where reparameterizing the parameter space using a non-linear transformation is proposed, the non-linear algorithm was derived. This method utilizes differential geometry and *a priori* knowledge of the system behaviour to develop the transformation. In this work, the three system rate functions were chosen as pseudo-outputs. In the non-linear algorithm, the method of characteristics is used to determine the inestimable parameter combinations in order to partition the parameter space into estimable and inestimable subspaces. Since only the pseudo-outputs need to be estimated, the quantities required to be estimated decreased from twelve to three. The three pseudo-parameters were found to produce a unique local minimum for the WSSE, however, the WSSE was found to have several local minima, implying that a global optimization procedure should be used in order to compute a global minimum. Similar to the linear algorithm, based on initial simulations, it can be seen that the non-linear model reparameterization algorithm, while achieving a good degree of output prediction, also greatly decreased the computational expense.

There are significant differences between the non-linear algorithm proposed in this work and the one proposed by Ben-Zvi (2008). First, in order to obtain an analytical solution, second-order Taylor approximations of the pseudo-outputs were used. Second, the transformation from the parameter space to the pseudo-parameter space was non-linear and the inverse transformation was obtained numerically. Finally, an explicit equation for computing the sensitivity of the output with respect to the pseudo-parameters at specific sampling times was realized.

Using simulations, a detailed comparison between the proposed linear and non-linear algorithms resulted in the non-linear algorithm having better performance, as it resulted in lower WSSE even with a shorter computational time.

Using the sensitivity of the system outputs with respect to the three pseudo-parameters, a D-optimal input profile was designed. This feed profile was implemented on an actual bioreactor system. Experimental data pertaining to the four system outputs were collected and analyzed. Implementation of the non-linear model reparameterization algorithm did not result in adequate degree of output prediction.

The non-linear model reparameterization algorithm was modified by introducing an iteration loop such that for every consequent iteration, the point of approximation is updated based on parameter estimates obtained in the previous iteration. This iterative non-linear algorithm is able to achieve a good degree of output prediction. When compared to the scenario where all twelve parameters were estimated, the iterative non-linear algorithm has a slightly (0.3%) higher WSSE; however, its computational time is 97% lower. Hence, the iterative non-linear algorithm can be part of a model-based strategies used for the purpose of control and optimization.

5.2 Future work

There is significant potential and scope in the field of modeling and control of a microalgal bioreactor. In order to better understand the dynamics of the microalgal bioreactor and improve the economic competiveness of producing biodiesel using microalgae, the following objectives for future work are suggested:

1. The algorithm proposed led to good output prediction; however, due to parameter correlations, the parameter estimated may not be entirely accurate. Therefore, there is still work to be done with regards to parameter estimation. Obtain additional measurements to improve the accuracy of parameter estimates.

2. Optimal experimental design can be used to find out the most suitable media formulation for oil productivity.

3. The microalgal bioreactor model should be modified so that it is more able to capture the transient dynamics of the reactor over a wide range of state values, should be investigated.

4. A model-based predictive control algorithm should be designed and implemented to take advantage of the model obtained in this work.

References

Ben-zvi, Amos. "Reparameterization of inestimable systems with applications to chemical and biochemical reactor systems." *AIChE Journal* 54, no. 5 (2008): 1270-1281.

Minowa, T, S Yokoyama, M Kishimoto, and T Okakura. "Oil Production from Algal Cells of *Dunaliella tertiolecta* by Direct Thermochemical Liquefaction." *Fuel* 74, no. 12 (1995): 1735-1738.

A

MATLAB® and MAPLE™ codes for the iterative non-linear algorithm

This appendix contains the MATLAB® and MAPLE™ codes for the iterative non-linear model reparameterization algorithm proposed in Chapter 5.

A.1 Main m-file

```
%=========================================================================
% File: RunningThreePrediction.m
% Description: This is the main file for the iterative non-linear
%               algorithm.

%=========================================================================

%-------------------------------------------------------------------------
%% Start with a clear memory
%-------------------------------------------------------------------------
clear
close all
clc

%-------------------------------------------------------------------------
%% Initialize the stopping condition
%-------------------------------------------------------------------------
CurrErr = 1000;
PreErr = 100000;
save('PreError.mat','PreErr')

%-------------------------------------------------------------------------
%% Load initial guess for tao (pseudo-parameters) and K (real parameters)
%-------------------------------------------------------------------------
load Taoold1.mat                        % Contains tao
load Kinitialguess.mat                  % Contains kold

Kold = Kguess;
save('Kold.mat','Kold');

load PatternSearchOptions.mat
% Contains psoptions3 and psoptions1 - Pattern search options

%-------------------------------------------------------------------------
%% Initialize counter and start timer
%-------------------------------------------------------------------------
z = 0;
```

```
tic

%--------------------------------------------------------------------------
%% A while loop to continue iterations until stopping condition is satisfied
%--------------------------------------------------------------------------
while CurrErr >= 0.005
    z = z+1

%--------------------------------------------------------------------------
%% Numerically solving for tao at each iteration
%--------------------------------------------------------------------------
    tao = patternsearch(@gettingTao,tao,[],[],[],[],[],[],[],psoptions3);
    save('Taoold.mat','tao');

%--------------------------------------------------------------------------
%% Solving for pseudo-parameters
%--------------------------------------------------------------------------
    t3 = tao(1:3,1);
    t3new =
patternsearch(@PredictionError,t3,[],[],[],[],[],[],[],psoptions1);
    save('CurrentTao3','t3new');

%--------------------------------------------------------------------------
%% Calculating Error and new real parameter values
%--------------------------------------------------------------------------
    Error = PredictionError(t3new);
    Kold = gettingK(t3new);
    save('Kold.mat','Kold');

%--------------------------------------------------------------------------
%% Calculating stopping condition
%--------------------------------------------------------------------------
    load PreError.mat
    CurrErr = abs(PreErr - Error);
    save('PreError.mat','PreErr')

%--------------------------------------------------------------------------
%% Saving all the results of the iteration
%--------------------------------------------------------------------------
    save('CurrentIteration.mat','tao','Kold','t3new','Error')

end
toc
```

A.2 Numerically solving for τ

```
%=========================================================================
% File: gettingTao.m
% Description: Numerically sovling for tao. This function is used by the
%               optimizer to find tao for a set of K.

%=========================================================================
function Error = gettingTao(tao)

load Kold.mat
load State1.mat
%-------------------------------------------------------------------------
%% Assigning States
%-------------------------------------------------------------------------
xnorm = State(1,1);
s1norm = State(2,1);
s2norm = State(3,1);
qnorm = State(4,1);
pnorm = State(5,1);
Vnorm = State(6,1);

%-------------------------------------------------------------------------
%% Assigning real parameters
%-------------------------------------------------------------------------
P1norm = Kold(1,1);
P2norm = Kold(2,1);
P3norm = Kold(3,1);
P4norm = Kold(4,1);
P5norm = Kold(5,1);
P6norm = Kold(6,1);
P7norm = Kold(7,1);
P8norm = Kold(8,1);
P9norm = Kold(9,1);
P10norm = Kold(10,1);
P11norm = Kold(11,1);
P12norm = Kold(12,1);

%-------------------------------------------------------------------------
%% Assigning Pseudo-parameters
%-------------------------------------------------------------------------
t1 = tao(1,1);
t2 = tao(2,1);
t3 = tao(3,1);
t4 = tao(4,1);
t5 = tao(5,1);
t6 = tao(6,1);
t7 = tao(7,1);
t8 = tao(8,1);
t9 = tao(9,1);
t10 = tao(10,1);
t11 = tao(11,1);
t12 = tao(12,1);

%-------------------------------------------------------------------------
%% Transformation
%-------------------------------------------------------------------------
Knew = .... The non-linear transformation is too long to be in the thesis.
% The non-linear transformation is acquired from the MAPLE™ code

%-------------------------------------------------------------------------
%% Calculating Error
%-------------------------------------------------------------------------
Error = (Knew-Kold)'*(Knew-Kold);
return;
```

A.3 Calculating real parameter values from pseudo-parameters

```
%=========================================================================
% File: gettingK.m
% Description: Calculating real parameter values from the three ....
%               pseudo-parameters from the non-linear algorithm, t3
%=========================================================================
function Knew = gettingK(t3)

load Taoold.mat
load Kold.mat

t = tao;
for i = 1:length(t3)
    t(i,1) = t3(i,1);
end

Knew = transformation(t);

return;
```

A.4 Calculating prediction error

```
%=========================================================================
% File: PredictionError.m
% Description: Calculates the prediction error based on the three
%               pseudo-parameters from the non-linear algorithm, t3.

%=========================================================================

function Err = PredictionError(t3)

%-------------------------------------------------------------------------
%% Structuring tao
%-------------------------------------------------------------------------
load Taoold.mat
% Load tao that was acquired from numerically solving for tao based on ...
% real parameter values from previous iteration

t = tao;
for i = 1:length(t3)
    t(i,1) = t3(i,1);
end
% t(1:3,1) are three pseudo-parameter values that are estimated while ...
% t(4:12,1) are three pseudo-parameter values that are not estimated

%-------------------------------------------------------------------------
%% Calculating real parameter values
%-------------------------------------------------------------------------
Knew = transformation(t,Kold);
% Calculating real parameter values based in each guess of the three ...
% pseudo-parameters from the non-linear algorithm

%-------------------------------------------------------------------------
%% Constraining tao based on real parameter constraints
%-------------------------------------------------------------------------
load HectorKvalues1.mat
% Contains upper (H) and lower (L) limits for the real parameters (K)

for i = 1:length(Knew)
    if Knew(i,1) >= L(i,1) & Knew(i,1) <= H(i,1)
        A(i,1) = 1;
    else
        A(i,1) = -1;
    end
end

B = sum(A);

%-------------------------------------------------------------------------
%% Calculating prediciton error
%-------------------------------------------------------------------------
if B == 12

    load GlycineFlowRate.mat        % Loading experimental data
    load glucoseFlowRate.mat        % Loading experimental data
    load FlowOut.mat                % Loading experimental data
    load Yavg.mat                   % Loading average output values


    X0 = [3.374e-4;4.3e-4;4.149e-2;2.25e-5;4.97e-4;1060]; %Initial state
conditions
    [T,X1] = simulateReactor2(f1,f2,f0,X0,Knew);
    y1 = X1(:,1) + X1(:,4) + X1(:,5);
    y2 = X1(:,2);
    y3 = X1(:,3);
    y4 = X1(:,5)./y1;
```

```
    Ypred = [y1 y2 y3 y4];

    Y1 = Ypred(:,1)/Y1avg;          % Weighing be inverse of the average value
    Y2 = Ypred(:,2)/Y2avg;          % Weighing be inverse of the average value
    Y3 = Ypred(:,3)/Y3avg;          % Weighing be inverse of the average value
    Y4 = Ypred(:,4)/Y4avg;          % Weighing be inverse of the average value

    load BiomassOuput.mat            % Loading experimental data
    load GlycineOutput.mat           % Loading experimental data
    load GlucoseOutput.mat           % Loading experimental data
    load OilOuput.mat                % Loading experimental data

    Y1exp(:,2) = Y1exp(:,2)/Y1avg; % Weighing be inverse of the average value
    Y2exp(:,2) = Y2exp(:,2)/Y2avg; % Weighing be inverse of the average value
    Y3exp(:,2) = Y3exp(:,2)/Y3avg; % Weighing be inverse of the average value
    Y4exp(:,2) = Y4exp(:,2)/Y4avg; % Weighing be inverse of the average value

    Y1pred = zeros(length(Y1exp),1);
    Y2pred = zeros(length(Y2exp),1);
    Y3pred = zeros(length(Y3exp),1);
    Y4pred = zeros(length(Y4exp),1);

 % Since samples are not taken every hour, the predicted output needs to...
 % be samples to match the timing of the experimental data.

    % Biomass
    a = length(Y1exp);
    for i = 1:a
        Tb = Y1exp(i,1);
        Y1pred(i,1) = Y1(Y1exp(i,1),1);
    end

    % Glycine
    a = length(Y2exp);
    for i = 1:a
        Tb = Y2exp(i,1);
        Y2pred(i,1) = Y2(Y2exp(i,1),1);
    end

    % Glucose
    a = length(Y3exp);
    for i = 1:a
        Tb = Y3exp(i,1);
        Y3pred(i,1) = Y3(Y3exp(i,1),1);
    end

    % Oil
    a = length(Y4exp);
    for i = 1:a
        Tb = Y4exp(i,1);
        Y4pred(i,1) = Y4(Y4exp(i,1),1);
    end

    WE1 = (Y1pred-Y1exp(:,2))'*(Y1pred-Y1exp(:,2));
    WE2 = (Y2pred-Y2exp(:,2))'*(Y2pred-Y2exp(:,2));
    WE3 = (Y3pred-Y3exp(:,2))'*(Y3pred-Y3exp(:,2));
    WE4 = (Y4pred-Y4exp(:,2))'*(Y4pred-Y4exp(:,2));

    WError = WE1+WE2+WE3+WE4;
    Err = WError;

else
   % if the real parameter constraints are violated, the error is assigned...
   % a very large value
    Err = 10000;
end
 return;
```

A.5 Non-linear transformation

```
%=========================================================================
% File: transformation.m
% Description: Converts pseudo-parameters to real parameters

%=========================================================================

function Knew = transformation(tao)

load State1.mat
load Kold.mat

%-------------------------------------------------------------------------
%% Assigning States
%-------------------------------------------------------------------------
xnorm = State(1,1);
s1norm = State(2,1);
s2norm = State(3,1);
qnorm = State(4,1);
pnorm = State(5,1);
Vnorm = State(6,1);


%-------------------------------------------------------------------------
%% Assigning real parameters
%-------------------------------------------------------------------------
P1norm = Kold(1,1);
P2norm = Kold(2,1);
P3norm = Kold(3,1);
P4norm = Kold(4,1);
P5norm = Kold(5,1);
P6norm = Kold(6,1);
P7norm = Kold(7,1);
P8norm = Kold(8,1);
P9norm = Kold(9,1);
P10norm = Kold(10,1);
P11norm = Kold(11,1);
P12norm = Kold(12,1);


%-------------------------------------------------------------------------
%% Assigning Pseudo-parameters
%-------------------------------------------------------------------------
t1 = tao(1,1);
t2 = tao(2,1);
t3 = tao(3,1);
t4 = tao(4,1);
t5 = tao(5,1);
t6 = tao(6,1);
t7 = tao(7,1);
t8 = tao(8,1);
t9 = tao(9,1);
t10 = tao(10,1);
t11 = tao(11,1);
t12 = tao(12,1);

%-------------------------------------------------------------------------
%% Transformation
%-------------------------------------------------------------------------
Knew = .... The non-linear transformation is too long to be in the thesis.
% The non-linear transformation is acquired from the MAPLE™ code

return;
```

A.6 Simulating the bioreactor model

```
%=========================================================================
% File: simulateReactor2.m
% Description: Simulating the bio-reactor model
% Inputs: f1 = Flowrate of nitrogen rich feed
%         f2 = flowrate of carbon rich feed
%         f0 = Outlet flowrate
%         X0 = Initial state conditions
%         K  = Model parameter values

% Outputs: T  = Time vector
%          X1 = State trajectories

%=========================================================================

function [T,X1] = simulateReactor2(f1,f2,f0,X0,K)

% Initialize state values
x1k(1,1) = X0(1,1);
x2k(1,1) = X0(2,1);
x3k(1,1) = X0(3,1);
x4k(1,1) = X0(4,1);
x5k(1,1) = X0(5,1);
x6k(1,1) = X0(6,1);
t(1,1) = 0;


for i=1:length(f1)

x1(i,1) = x1k(length(x1k),1);
x2(i,1) = x2k(length(x2k),1);
x3(i,1) = x3k(length(x3k),1);
x4(i,1) = x4k(length(x4k),1);
x5(i,1) = x5k(length(x5k),1);
x6(i,1) = x6k(length(x6k),1);
T2(i,1) = t(length(t),1);


   x1i=x1k(length(x1k),1);
   x2i=x2k(length(x2k),1);
   x3i=x3k(length(x3k),1);
   x4i=x4k(length(x4k),1);
   x5i=x5k(length(x5k),1);
   x6i=x6k(length(x6k),1);
   xi = [x1i;x2i;x3i;x4i;x5i;x6i];

  [T1,W]=ode45(@(t,W) biomNewModel(t,W,f1(i,1),f2(i,1),f0(i,1),K),i-
1:1:i,xi);


   t=[t;T1];
   x1k=[x1;W(:,1)];
   x2k=[x2;W(:,2)];
   x3k=[x3;W(:,3)];
   x4k=[x4;W(:,4)];
   x5k=[x5;W(:,5)];
   x6k=[x6;W(:,6)];

end

X1 = [x1 x2 x3 x4 x5 x6];
T = T2;

return;
```

A.7 Bioreactor model

```
%=========================================================================
% File: biomNewModel.m
% Description: Bioreactor model
% Inputs: f1 = Flowrate of nitrogen rich feed
%         f2 = Flowrate of carbon rich feed
%         f0 = Outlet flowrate
%         W = State trajectories
%         t  = Time vector

% Outputs: R  = State trajectories
%=========================================================================
function R = biomNewModel(t,W,f1,f2,f0,K)

% States
x = W(1,1);
s1 = W(2,1);
s2 = W(3,1);
q = W(4,1);
p = W(5,1);
V = W(6,1);

% Inputs
fi1 = f1;
fi2 = f2;
fo = f0;

% Feed concentrations
s1i = 0.01;
s2i = 0.2;

R = zeros(6,1);

% Parameter values
P1 = K(1,1);
P2 = K(2,1);
P3 = K(3,1);
P4 = K(4,1);
P5 = K(5,1);
P6 = K(6,1);
P7 = K(7,1);
P8 = K(8,1);
P9 = K(9,1);
P10 = K(10,1);
P11 = K(11,1);
P12 = K(12,1);

e = 0.001;

% Rates functions
qr = q/(x+q+p);
pr = p/(x+q+p);

if s2 == 0
    mu = 0;
else
    mu = max(0,P1 * (qr-P2)/(P3+qr) * s2/(P5+s2));
end

if s1>P6
rho = P4 * (1-(P6/s1))^(1+e);
else
    rho = 0;
end
```

```
pik = P10 * s2/(P11+s2) * (1-pr);



% Steady state equations
R(1,1) = (mu*x)-(x*fo/V)-(x*(fi1+fi2-fo)/V);
R(2,1) = -rho*x+s1i*fi1/V-s1*fo/V-s1*(fi1+fi2-fo)/V;
R(3,1) = -P7*mu*x+s2i*fi2/V-s2*fo/V-P8*x-P9*pik*x-s2*(fi1+fi2-fo)/V;
R(4,1) = rho*x-P12*mu*x-q*fo/V-q*(fi1+fi2-fo)/V;
R(5,1) = pik*x-p*fo/V-p*(fi1+fi2-fo)/V;
R(6,1) = fi1+fi2-fo;

return;
```

A.8 MAPLE™ code for getting the non-linear transformation

```
> restart:
> with(linalg):
>
> with(Matlab):
> with(inttrans):
>
> mu:=P1*(q-P2)*s2/((P3+q-P2)*(1+s2)):
> rho:=P4*s1*(1-P6/s1)/(P5+s1-P6):
> pi:=P10*s2/(P11+s2):
>
> P_1_plus:=mu:
> P_2_plus:=rho:
> P_3_plus:=pi:
> params := [P1, P2, P3, P4, P5, P6, P7, P8, P9, P10, P11, P12]:
>
> Para_set := {P1=P1norm,
P2=P2norm,P3=P3norm,P4=P4norm,P5=P5norm,P6=P6norm,P7=P7norm,P8=P8norm,P9=P9norm,P1
0=P10norm,P11=P11norm,P12=P12norm}:
>
> Para_nom :=
[P1norm,P2norm,P3norm,P4norm,P5norm,P6norm,P7norm,P8norm,P9norm,P10norm,P11norm,P1
2norm]:
> P_sub := params-Para_nom:
>
> State_set:= {x=xnorm,s1=s1norm,s2=s2norm,q=qnorm,p=pnorm,V=Vnorm}:
>
> P_1:= [P_1_plus, P_2_plus,P_3_plus]:
> P:=subs(State_set, evalm(P_1)):
> d_phi :=jacobian(P,params):
>
> d_phi_1_1k:=d_phi[1,1]:
> f_1_1k:=subs(Para_set, evalm(d_phi_1_1k)):
> dfdp_1_1k:=grad(d_phi_1_1k,params):
> dfdp_1_1k:=subs(Para_set,evalm(dfdp_1_1k)):
> sim_dfdp_1_1k:=evalm(evalm(dfdp_1_1k).P_sub)+f_1_1k:
> B_1_1k:=matrix(1,12,[evalm(coeff(sim_dfdp_1_1k, P1,
1)),evalm(coeff(sim_dfdp_1_1k, P2, 1)),evalm(coeff(sim_dfdp_1_1k, P3,
1)),evalm(coeff(sim_dfdp_1_1k, P4, 1)),evalm(coeff(sim_dfdp_1_1k, P5,
1)),evalm(coeff(sim_dfdp_1_1k, P6, 1)),evalm(coeff(sim_dfdp_1_1k, P7,
1)),evalm(coeff(sim_dfdp_1_1k, P8, 1)),evalm(coeff(sim_dfdp_1_1k, P9,
1)),evalm(coeff(sim_dfdp_1_1k, P10, 1)),evalm(coeff(sim_dfdp_1_1k, P11,
1)),evalm(coeff(sim_dfdp_1_1k, P12, 1))]):
> tem:= evalm(B_1_1k.params):
> A_1_1k:=matrix(1,1,[evalm(sim_dfdp_1_1k-tem)]):
>
> d_phi_1_2k:=d_phi[1,2]:
> f_1_2k:=subs(Para_set, evalm(d_phi_1_2k)):
> dfdp_1_2k:=grad(d_phi_1_2k,params):
> dfdp_1_2k:=subs(Para_set,evalm(dfdp_1_2k)):
> sim_dfdp_1_2k:=evalm(evalm(dfdp_1_2k).P_sub)+f_1_2k:
> B_1_2k:=matrix(1,12,[evalm(coeff(sim_dfdp_1_2k, P1,
1)),evalm(coeff(sim_dfdp_1_2k, P2, 1)),evalm(coeff(sim_dfdp_1_2k, P3,
1)),evalm(coeff(sim_dfdp_1_2k, P4, 1)),evalm(coeff(sim_dfdp_1_2k, P5,
1)),evalm(coeff(sim_dfdp_1_2k, P6, 1)),evalm(coeff(sim_dfdp_1_2k, P7,
1)),evalm(coeff(sim_dfdp_1_2k, P8, 1)),evalm(coeff(sim_dfdp_1_2k, P9,
1)),evalm(coeff(sim_dfdp_1_2k, P10, 1)),evalm(coeff(sim_dfdp_1_2k, P11,
1)),evalm(coeff(sim_dfdp_1_2k, P12, 1))]):
> tem:= evalm(B_1_2k.params):
> A_1_2k:=matrix(1,1,[evalm(sim_dfdp_1_2k-tem)]):
>
> d_phi_1_3k:=d_phi[1,3]:
> f_1_3k:=subs(Para_set, evalm(d_phi_1_3k)):
> dfdp_1_3k:=grad(d_phi_1_3k,params):
> dfdp_1_3k:=subs(Para_set,evalm(dfdp_1_3k)):
> sim_dfdp_1_3k:=evalm(evalm(dfdp_1_3k).P_sub)+f_1_3k:
> B_1_3k:=matrix(1,12,[evalm(coeff(sim_dfdp_1_3k, P1,
1)),evalm(coeff(sim_dfdp_1_3k, P2, 1)),evalm(coeff(sim_dfdp_1_3k, P3,
1)),evalm(coeff(sim_dfdp_1_3k, P4, 1)),evalm(coeff(sim_dfdp_1_3k, P5,
```

```
1)),evalm(coeff(sim_dfdp_1_3k, P6, 1)),evalm(coeff(sim_dfdp_1_3k, P7,
1)),evalm(coeff(sim_dfdp_1_3k, P8, 1)),evalm(coeff(sim_dfdp_1_3k, P9,
1)),evalm(coeff(sim_dfdp_1_3k, P10, 1)),evalm(coeff(sim_dfdp_1_3k, P11,
1)),evalm(coeff(sim_dfdp_1_3k, P12, 1))]):
> tem:= evalm(B_1_3k.params):
> A_1_3k:=matrix(1,1,[evalm(sim_dfdp_1_3k-tem)]):
>
> d_phi_1_4k:=d_phi[1,4]:
> f_1_4k:=subs(Para_set, evalm(d_phi_1_4k)):
> dfdp_1_4k:=grad(d_phi_1_4k,params):
> dfdp_1_4k:=subs(Para_set,evalm(dfdp_1_4k)):
> sim_dfdp_1_4k:=evalm(evalm(dfdp_1_4k).P_sub)+f_1_4k:
> B_1_4k:=matrix(1,12,[evalm(coeff(sim_dfdp_1_4k, P1,
1)),evalm(coeff(sim_dfdp_1_4k, P2, 1)),evalm(coeff(sim_dfdp_1_4k, P3,
1)),evalm(coeff(sim_dfdp_1_4k, P4, 1)),evalm(coeff(sim_dfdp_1_4k, P5,
1)),evalm(coeff(sim_dfdp_1_4k, P6, 1)),evalm(coeff(sim_dfdp_1_4k, P7,
1)),evalm(coeff(sim_dfdp_1_4k, P8, 1)),evalm(coeff(sim_dfdp_1_4k, P9,
1)),evalm(coeff(sim_dfdp_1_4k, P10, 1)),evalm(coeff(sim_dfdp_1_4k, P11,
1)),evalm(coeff(sim_dfdp_1_4k, P12, 1))]):
> tem:= evalm(B_1_4k.params):
> A_1_4k:=matrix(1,1,[evalm(sim_dfdp_1_4k-tem)]):
>
> d_phi_1_5k:=d_phi[1,5]:
> f_1_5k:=subs(Para_set, evalm(d_phi_1_5k)):
> dfdp_1_5k:=grad(d_phi_1_5k,params):
> dfdp_1_5k:=subs(Para_set,evalm(dfdp_1_5k)):
> sim_dfdp_1_5k:=evalm(evalm(dfdp_1_5k).P_sub)+f_1_5k:
> B_1_5k:=matrix(1,12,[evalm(coeff(sim_dfdp_1_5k, P1,
1)),evalm(coeff(sim_dfdp_1_5k, P2, 1)),evalm(coeff(sim_dfdp_1_5k, P3,
1)),evalm(coeff(sim_dfdp_1_5k, P4, 1)),evalm(coeff(sim_dfdp_1_5k, P5,
1)),evalm(coeff(sim_dfdp_1_5k, P6, 1)),evalm(coeff(sim_dfdp_1_5k, P7,
1)),evalm(coeff(sim_dfdp_1_5k, P8, 1)),evalm(coeff(sim_dfdp_1_5k, P9,
1)),evalm(coeff(sim_dfdp_1_5k, P10, 1)),evalm(coeff(sim_dfdp_1_5k, P11,
1)),evalm(coeff(sim_dfdp_1_5k, P12, 1))]):
> tem:= evalm(B_1_5k.params):
> A_1_5k:=matrix(1,1,[evalm(sim_dfdp_1_5k-tem)]):
>
> d_phi_1_6k:=d_phi[1,6]:
> f_1_6k:=subs(Para_set, evalm(d_phi_1_6k)):
> dfdp_1_6k:=grad(d_phi_1_6k,params):
> dfdp_1_6k:=subs(Para_set,evalm(dfdp_1_6k)):
> sim_dfdp_1_6k:=evalm(evalm(dfdp_1_6k).P_sub)+f_1_6k:
> B_1_6k:=matrix(1,12,[evalm(coeff(sim_dfdp_1_6k, P1,
1)),evalm(coeff(sim_dfdp_1_6k, P2, 1)),evalm(coeff(sim_dfdp_1_6k, P3,
1)),evalm(coeff(sim_dfdp_1_6k, P4, 1)),evalm(coeff(sim_dfdp_1_6k, P5,
1)),evalm(coeff(sim_dfdp_1_6k, P6, 1)),evalm(coeff(sim_dfdp_1_6k, P7,
1)),evalm(coeff(sim_dfdp_1_6k, P8, 1)),evalm(coeff(sim_dfdp_1_6k, P9,
1)),evalm(coeff(sim_dfdp_1_6k, P10, 1)),evalm(coeff(sim_dfdp_1_6k, P11,
1)),evalm(coeff(sim_dfdp_1_6k, P12, 1))]):
> tem:= evalm(B_1_6k.params):
> A_1_6k:=matrix(1,1,[evalm(sim_dfdp_1_6k-tem)]):
>
> d_phi_1_7k:=d_phi[1,7]:
> f_1_7k:=subs(Para_set, evalm(d_phi_1_7k)):
> dfdp_1_7k:=grad(d_phi_1_7k,params):
> dfdp_1_7k:=subs(Para_set,evalm(dfdp_1_7k)):
> sim_dfdp_1_7k:=evalm(evalm(dfdp_1_7k).P_sub)+f_1_7k:
> B_1_7k:=matrix(1,12,[evalm(coeff(sim_dfdp_1_7k, P1,
1)),evalm(coeff(sim_dfdp_1_7k, P2, 1)),evalm(coeff(sim_dfdp_1_7k, P3,
1)),evalm(coeff(sim_dfdp_1_7k, P4, 1)),evalm(coeff(sim_dfdp_1_7k, P5,
1)),evalm(coeff(sim_dfdp_1_7k, P6, 1)),evalm(coeff(sim_dfdp_1_7k, P7,
1)),evalm(coeff(sim_dfdp_1_7k, P8, 1)),evalm(coeff(sim_dfdp_1_7k, P9,
1)),evalm(coeff(sim_dfdp_1_7k, P10, 1)),evalm(coeff(sim_dfdp_1_7k, P11,
1)),evalm(coeff(sim_dfdp_1_7k, P12, 1))]):
> tem:= evalm(B_1_7k.params):
> A_1_7k:=matrix(1,1,[evalm(sim_dfdp_1_7k-tem)]):
>
> d_phi_1_8k:=d_phi[1,8]:
> f_1_8k:=subs(Para_set, evalm(d_phi_1_8k)):
> dfdp_1_8k:=grad(d_phi_1_8k,params):
> dfdp_1_8k:=subs(Para_set,evalm(dfdp_1_8k)):
```

```
> sim_dfdp_1_8k:=evalm(evalm(dfdp_1_8k).P_sub)+f_1_8k:
> B_1_8k:=matrix(1,12,[evalm(coeff(sim_dfdp_1_8k, P1,
1)),evalm(coeff(sim_dfdp_1_8k, P2, 1)),evalm(coeff(sim_dfdp_1_8k, P3,
1)),evalm(coeff(sim_dfdp_1_8k, P4, 1)),evalm(coeff(sim_dfdp_1_8k, P5,
1)),evalm(coeff(sim_dfdp_1_8k, P6, 1)),evalm(coeff(sim_dfdp_1_8k, P7,
1)),evalm(coeff(sim_dfdp_1_8k, P8, 1)),evalm(coeff(sim_dfdp_1_8k, P9,
1)),evalm(coeff(sim_dfdp_1_8k, P10, 1)),evalm(coeff(sim_dfdp_1_8k, P11,
1)),evalm(coeff(sim_dfdp_1_8k, P12, 1))]):
> tem:= evalm(B_1_8k.params):
> A_1_8k:=matrix(1,1,[evalm(sim_dfdp_1_8k-tem)]):
>
> d_phi_1_9k:=d_phi[1,9]:
> f_1_9k:=subs(Para_set, evalm(d_phi_1_9k)):
> dfdp_1_9k:=grad(d_phi_1_9k,params):
> dfdp_1_9k:=subs(Para_set,evalm(dfdp_1_9k)):
> sim_dfdp_1_9k:=evalm(evalm(dfdp_1_9k).P_sub)+f_1_9k:
> B_1_9k:=matrix(1,12,[evalm(coeff(sim_dfdp_1_9k, P1,
1)),evalm(coeff(sim_dfdp_1_9k, P2, 1)),evalm(coeff(sim_dfdp_1_9k, P3,
1)),evalm(coeff(sim_dfdp_1_9k, P4, 1)),evalm(coeff(sim_dfdp_1_9k, P5,
1)),evalm(coeff(sim_dfdp_1_9k, P6, 1)),evalm(coeff(sim_dfdp_1_9k, P7,
1)),evalm(coeff(sim_dfdp_1_9k, P8, 1)),evalm(coeff(sim_dfdp_1_9k, P9,
1)),evalm(coeff(sim_dfdp_1_9k, P10, 1)),evalm(coeff(sim_dfdp_1_9k, P11,
1)),evalm(coeff(sim_dfdp_1_9k, P12, 1))]):
> tem:= evalm(B_1_9k.params):
> A_1_9k:=matrix(1,1,[evalm(sim_dfdp_1_9k-tem)]):
>
> d_phi_1_10k:=d_phi[1,10]:
> f_1_10k:=subs(Para_set, evalm(d_phi_1_10k)):
> dfdp_1_10k:=grad(d_phi_1_10k,params):
> dfdp_1_10k:=subs(Para_set,evalm(dfdp_1_10k)):
> sim_dfdp_1_10k:=evalm(evalm(dfdp_1_10k).P_sub)+f_1_10k:
> B_1_10k:=matrix(1,12,[evalm(coeff(sim_dfdp_1_10k, P1,
1)),evalm(coeff(sim_dfdp_1_10k, P2, 1)),evalm(coeff(sim_dfdp_1_10k, P3,
1)),evalm(coeff(sim_dfdp_1_10k, P4, 1)),evalm(coeff(sim_dfdp_1_10k, P5,
1)),evalm(coeff(sim_dfdp_1_10k, P6, 1)),evalm(coeff(sim_dfdp_1_10k, P7,
1)),evalm(coeff(sim_dfdp_1_10k, P8, 1)),evalm(coeff(sim_dfdp_1_10k, P9,
1)),evalm(coeff(sim_dfdp_1_10k, P10, 1)),evalm(coeff(sim_dfdp_1_10k, P11,
1)),evalm(coeff(sim_dfdp_1_10k, P12, 1))]):
> tem:= evalm(B_1_10k.params):
> A_1_10k:=matrix(1,1,[evalm(sim_dfdp_1_10k-tem)]):
>
> d_phi_1_11k:=d_phi[1,11]:
> f_1_11k:=subs(Para_set, evalm(d_phi_1_11k)):
> dfdp_1_11k:=grad(d_phi_1_11k,params):
> dfdp_1_11k:=subs(Para_set,evalm(dfdp_1_11k)):
> sim_dfdp_1_11k:=evalm(evalm(dfdp_1_11k).P_sub)+f_1_11k:
> B_1_11k:=matrix(1,12,[evalm(coeff(sim_dfdp_1_11k, P1,
1)),evalm(coeff(sim_dfdp_1_11k, P2, 1)),evalm(coeff(sim_dfdp_1_11k, P3,
1)),evalm(coeff(sim_dfdp_1_11k, P4, 1)),evalm(coeff(sim_dfdp_1_11k, P5,
1)),evalm(coeff(sim_dfdp_1_11k, P6, 1)),evalm(coeff(sim_dfdp_1_11k, P7,
1)),evalm(coeff(sim_dfdp_1_11k, P8, 1)),evalm(coeff(sim_dfdp_1_11k, P9,
1)),evalm(coeff(sim_dfdp_1_11k, P10, 1)),evalm(coeff(sim_dfdp_1_11k, P11,
1)),evalm(coeff(sim_dfdp_1_11k, P12, 1))]):
> tem:= evalm(B_1_11k.params):
> A_1_11k:=matrix(1,1,[evalm(sim_dfdp_1_11k-tem)]):
>
> d_phi_1_12k:=d_phi[1,12]:
> f_1_12k:=subs(Para_set, evalm(d_phi_1_12k)):
> dfdp_1_12k:=grad(d_phi_1_12k,params):
> dfdp_1_12k:=subs(Para_set,evalm(dfdp_1_12k)):
> sim_dfdp_1_12k:=evalm(evalm(dfdp_1_12k).P_sub)+f_1_12k:
> B_1_12k:=matrix(1,12,[evalm(coeff(sim_dfdp_1_12k, P1,
1)),evalm(coeff(sim_dfdp_1_12k, P2, 1)),evalm(coeff(sim_dfdp_1_12k, P3,
1)),evalm(coeff(sim_dfdp_1_12k, P4, 1)),evalm(coeff(sim_dfdp_1_12k, P5,
1)),evalm(coeff(sim_dfdp_1_12k, P6, 1)),evalm(coeff(sim_dfdp_1_12k, P7,
1)),evalm(coeff(sim_dfdp_1_12k, P8, 1)),evalm(coeff(sim_dfdp_1_12k, P9,
1)),evalm(coeff(sim_dfdp_1_12k, P10, 1)),evalm(coeff(sim_dfdp_1_12k, P11,
1)),evalm(coeff(sim_dfdp_1_12k, P12, 1))]):
> tem:= evalm(B_1_12k.params):
> A_1_12k:=matrix(1,1,[evalm(sim_dfdp_1_12k-tem)]):
>
```

```
>
A1:=stackmatrix(A_1_1k,A_1_2k,A_1_3k,A_1_4k,A_1_5k,A_1_6k,A_1_7k,A_1_8k,A_1_9k,A_1
_10k,A_1_11k,A_1_12k):
>
B1:=stackmatrix(B_1_1k,B_1_2k,B_1_3k,B_1_4k,B_1_5k,B_1_6k,B_1_7k,B_1_8k,B_1_9k,B_1
_10k,B_1_11k,B_1_12k):
> P_nom:=matrix(12,1,Para_nom):
> As1:=evalm(evalm(A1/s)-P_nom):
> T1:=evalm((diag(s,s,s,s,s,s,s,s,s,s,s,s)-B1)):
> Tinv1:=evalm(inverse(T1)):
> PL1:=evalm(Tinv1&*As1):
> pT1:=map(invlaplace,evalm(PL1),s,t1):
>
> d_phi_2_1k:=d_phi[2,1]:
> f_2_1k:=subs(Para_set, evalm(d_phi_2_1k)):
> dfdp_2_1k:=grad(d_phi_2_1k,params):
> dfdp_2_1k:=subs(Para_set,evalm(dfdp_2_1k)):
> sim_dfdp_2_1k:=evalm(evalm(dfdp_2_1k).P_sub)+f_2_1k:
> B_2_1k:=matrix(1,12,[evalm(coeff(sim_dfdp_2_1k, P1,
1)),evalm(coeff(sim_dfdp_2_1k, P2, 1)),evalm(coeff(sim_dfdp_2_1k, P3,
1)),evalm(coeff(sim_dfdp_2_1k, P4, 1)),evalm(coeff(sim_dfdp_2_1k, P5,
1)),evalm(coeff(sim_dfdp_2_1k, P6, 1)),evalm(coeff(sim_dfdp_2_1k, P7,
1)),evalm(coeff(sim_dfdp_2_1k, P8, 1)),evalm(coeff(sim_dfdp_2_1k, P9,
1)),evalm(coeff(sim_dfdp_2_1k, P10, 1)),evalm(coeff(sim_dfdp_2_1k, P11,
1)),evalm(coeff(sim_dfdp_2_1k, P12, 1))]):
> tem:= evalm(B_2_1k.params):
> A_2_1k:=matrix(1,1,[evalm(sim_dfdp_2_1k-tem)]):
>
> d_phi_2_2k:=d_phi[2,2]:
> f_2_2k:=subs(Para_set, evalm(d_phi_2_2k)):
> dfdp_2_2k:=grad(d_phi_2_2k,params):
> dfdp_2_2k:=subs(Para_set,evalm(dfdp_2_2k)):
> sim_dfdp_2_2k:=evalm(evalm(dfdp_2_2k).P_sub)+f_2_2k:
> B_2_2k:=matrix(1,12,[evalm(coeff(sim_dfdp_2_2k, P1,
1)),evalm(coeff(sim_dfdp_2_2k, P2, 1)),evalm(coeff(sim_dfdp_2_2k, P3,
1)),evalm(coeff(sim_dfdp_2_2k, P4, 1)),evalm(coeff(sim_dfdp_2_2k, P5,
1)),evalm(coeff(sim_dfdp_2_2k, P6, 1)),evalm(coeff(sim_dfdp_2_2k, P7,
1)),evalm(coeff(sim_dfdp_2_2k, P8, 1)),evalm(coeff(sim_dfdp_2_2k, P9,
1)),evalm(coeff(sim_dfdp_2_2k, P10, 1)),evalm(coeff(sim_dfdp_2_2k, P11,
1)),evalm(coeff(sim_dfdp_2_2k, P12, 1))]):
> tem:= evalm(B_2_2k.params):
> A_2_2k:=matrix(1,1,[evalm(sim_dfdp_2_2k-tem)]):
>
> d_phi_2_3k:=d_phi[2,3]:
> f_2_3k:=subs(Para_set, evalm(d_phi_2_3k)):
> dfdp_2_3k:=grad(d_phi_2_3k,params):
> dfdp_2_3k:=subs(Para_set,evalm(dfdp_2_3k)):
> sim_dfdp_2_3k:=evalm(evalm(dfdp_2_3k).P_sub)+f_2_3k:
> B_2_3k:=matrix(1,12,[evalm(coeff(sim_dfdp_2_3k, P1,
1)),evalm(coeff(sim_dfdp_2_3k, P2, 1)),evalm(coeff(sim_dfdp_2_3k, P3,
1)),evalm(coeff(sim_dfdp_2_3k, P4, 1)),evalm(coeff(sim_dfdp_2_3k, P5,
1)),evalm(coeff(sim_dfdp_2_3k, P6, 1)),evalm(coeff(sim_dfdp_2_3k, P7,
1)),evalm(coeff(sim_dfdp_2_3k, P8, 1)),evalm(coeff(sim_dfdp_2_3k, P9,
1)),evalm(coeff(sim_dfdp_2_3k, P10, 1)),evalm(coeff(sim_dfdp_2_3k, P11,
1)),evalm(coeff(sim_dfdp_2_3k, P12, 1))]):
> tem:= evalm(B_2_3k.params):
> A_2_3k:=matrix(1,1,[evalm(sim_dfdp_2_3k-tem)]):
>
> d_phi_2_4k:=d_phi[2,4]:
> f_2_4k:=subs(Para_set, evalm(d_phi_2_4k)):
> dfdp_2_4k:=grad(d_phi_2_4k,params):
> dfdp_2_4k:=subs(Para_set,evalm(dfdp_2_4k)):
> sim_dfdp_2_4k:=evalm(evalm(dfdp_2_4k).P_sub)+f_2_4k:
> B_2_4k:=matrix(1,12,[evalm(coeff(sim_dfdp_2_4k, P1,
1)),evalm(coeff(sim_dfdp_2_4k, P2, 1)),evalm(coeff(sim_dfdp_2_4k, P3,
1)),evalm(coeff(sim_dfdp_2_4k, P4, 1)),evalm(coeff(sim_dfdp_2_4k, P5,
1)),evalm(coeff(sim_dfdp_2_4k, P6, 1)),evalm(coeff(sim_dfdp_2_4k, P7,
1)),evalm(coeff(sim_dfdp_2_4k, P8, 1)),evalm(coeff(sim_dfdp_2_4k, P9,
1)),evalm(coeff(sim_dfdp_2_4k, P10, 1)),evalm(coeff(sim_dfdp_2_4k, P11,
1)),evalm(coeff(sim_dfdp_2_4k, P12, 1))]):
> tem:= evalm(B_2_4k.params):
```

```
> A_2_4k:=matrix(1,1,[evalm(sim_dfdp_2_4k-tem)]):
>
> d_phi_2_5k:=d_phi[2,5]:
> f_2_5k:=subs(Para_set, evalm(d_phi_2_5k)):
> dfdp_2_5k:=grad(d_phi_2_5k,params):
> dfdp_2_5k:=subs(Para_set,evalm(dfdp_2_5k)):
> sim_dfdp_2_5k:=evalm(evalm(dfdp_2_5k).P_sub)+f_2_5k:
> B_2_5k:=matrix(1,12,[evalm(coeff(sim_dfdp_2_5k, P1,
1)),evalm(coeff(sim_dfdp_2_5k, P2, 1)),evalm(coeff(sim_dfdp_2_5k, P3,
1)),evalm(coeff(sim_dfdp_2_5k, P4, 1)),evalm(coeff(sim_dfdp_2_5k, P5,
1)),evalm(coeff(sim_dfdp_2_5k, P6, 1)),evalm(coeff(sim_dfdp_2_5k, P7,
1)),evalm(coeff(sim_dfdp_2_5k, P8, 1)),evalm(coeff(sim_dfdp_2_5k, P9,
1)),evalm(coeff(sim_dfdp_2_5k, P10, 1)),evalm(coeff(sim_dfdp_2_5k, P11,
1)),evalm(coeff(sim_dfdp_2_5k, P12, 1))]):
> tem:= evalm(B_2_5k.params):
> A_2_5k:=matrix(1,1,[evalm(sim_dfdp_2_5k-tem)]):
>
> d_phi_2_6k:=d_phi[2,6]:
> f_2_6k:=subs(Para_set, evalm(d_phi_2_6k)):
> dfdp_2_6k:=grad(d_phi_2_6k,params):
> dfdp_2_6k:=subs(Para_set,evalm(dfdp_2_6k)):
> sim_dfdp_2_6k:=evalm(evalm(dfdp_2_6k).P_sub)+f_2_6k:
> B_2_6k:=matrix(1,12,[evalm(coeff(sim_dfdp_2_6k, P1,
1)),evalm(coeff(sim_dfdp_2_6k, P2, 1)),evalm(coeff(sim_dfdp_2_6k, P3,
1)),evalm(coeff(sim_dfdp_2_6k, P4, 1)),evalm(coeff(sim_dfdp_2_6k, P5,
1)),evalm(coeff(sim_dfdp_2_6k, P6, 1)),evalm(coeff(sim_dfdp_2_6k, P7,
1)),evalm(coeff(sim_dfdp_2_6k, P8, 1)),evalm(coeff(sim_dfdp_2_6k, P9,
1)),evalm(coeff(sim_dfdp_2_6k, P10, 1)),evalm(coeff(sim_dfdp_2_6k, P11,
1)),evalm(coeff(sim_dfdp_2_6k, P12, 1))]):
> tem:= evalm(B_2_6k.params):
> A_2_6k:=matrix(1,1,[evalm(sim_dfdp_2_6k-tem)]):
>
> d_phi_2_7k:=d_phi[2,7]:
> f_2_7k:=subs(Para_set, evalm(d_phi_2_7k)):
> dfdp_2_7k:=grad(d_phi_2_7k,params):
> dfdp_2_7k:=subs(Para_set,evalm(dfdp_2_7k)):
> sim_dfdp_2_7k:=evalm(evalm(dfdp_2_7k).P_sub)+f_2_7k:
> B_2_7k:=matrix(1,12,[evalm(coeff(sim_dfdp_2_7k, P1,
1)),evalm(coeff(sim_dfdp_2_7k, P2, 1)),evalm(coeff(sim_dfdp_2_7k, P3,
1)),evalm(coeff(sim_dfdp_2_7k, P4, 1)),evalm(coeff(sim_dfdp_2_7k, P5,
1)),evalm(coeff(sim_dfdp_2_7k, P6, 1)),evalm(coeff(sim_dfdp_2_7k, P7,
1)),evalm(coeff(sim_dfdp_2_7k, P8, 1)),evalm(coeff(sim_dfdp_2_7k, P9,
1)),evalm(coeff(sim_dfdp_2_7k, P10, 1)),evalm(coeff(sim_dfdp_2_7k, P11,
1)),evalm(coeff(sim_dfdp_2_7k, P12, 1))]):
> tem:= evalm(B_2_7k.params):
> A_2_7k:=matrix(1,1,[evalm(sim_dfdp_2_7k-tem)]):
>
> d_phi_2_8k:=d_phi[2,8]:
> f_2_8k:=subs(Para_set, evalm(d_phi_2_8k)):
> dfdp_2_8k:=grad(d_phi_2_8k,params):
> dfdp_2_8k:=subs(Para_set,evalm(dfdp_2_8k)):
> sim_dfdp_2_8k:=evalm(evalm(dfdp_2_8k).P_sub)+f_2_8k:
> B_2_8k:=matrix(1,12,[evalm(coeff(sim_dfdp_2_8k, P1,
1)),evalm(coeff(sim_dfdp_2_8k, P2, 1)),evalm(coeff(sim_dfdp_2_8k, P3,
1)),evalm(coeff(sim_dfdp_2_8k, P4, 1)),evalm(coeff(sim_dfdp_2_8k, P5,
1)),evalm(coeff(sim_dfdp_2_8k, P6, 1)),evalm(coeff(sim_dfdp_2_8k, P7,
1)),evalm(coeff(sim_dfdp_2_8k, P8, 1)),evalm(coeff(sim_dfdp_2_8k, P9,
1)),evalm(coeff(sim_dfdp_2_8k, P10, 1)),evalm(coeff(sim_dfdp_2_8k, P11,
1)),evalm(coeff(sim_dfdp_2_8k, P12, 1))]):
> tem:= evalm(B_2_8k.params):
> A_2_8k:=matrix(1,1,[evalm(sim_dfdp_2_8k-tem)]):
>
> d_phi_2_9k:=d_phi[2,9]:
> f_2_9k:=subs(Para_set, evalm(d_phi_2_9k)):
> dfdp_2_9k:=grad(d_phi_2_9k,params):
> dfdp_2_9k:=subs(Para_set,evalm(dfdp_2_9k)):
> sim_dfdp_2_9k:=evalm(evalm(dfdp_2_9k).P_sub)+f_2_9k:
> B_2_9k:=matrix(1,12,[evalm(coeff(sim_dfdp_2_9k, P1,
1)),evalm(coeff(sim_dfdp_2_9k, P2, 1)),evalm(coeff(sim_dfdp_2_9k, P3,
1)),evalm(coeff(sim_dfdp_2_9k, P4, 1)),evalm(coeff(sim_dfdp_2_9k, P5,
1)),evalm(coeff(sim_dfdp_2_9k, P6, 1)),evalm(coeff(sim_dfdp_2_9k, P7,
```

```
1)),evalm(coeff(sim_dfdp_2_9k, P8, 1)),evalm(coeff(sim_dfdp_2_9k, P9,
1)),evalm(coeff(sim_dfdp_2_9k, P10, 1)),evalm(coeff(sim_dfdp_2_9k, P11,
1)),evalm(coeff(sim_dfdp_2_9k, P12, 1))]):
> tem:= evalm(B_2_9k.params):
> A_2_9k:=matrix(1,1,[evalm(sim_dfdp_2_9k-tem)]):
>
> d_phi_2_10k:=d_phi[2,10]:
> f_2_10k:=subs(Para_set, evalm(d_phi_2_10k)):
> dfdp_2_10k:=grad(d_phi_2_10k,params):
> dfdp_2_10k:=subs(Para_set,evalm(dfdp_2_10k)):
> sim_dfdp_2_10k:=evalm(evalm(dfdp_2_10k).P_sub)+f_2_10k:
> B_2_10k:=matrix(1,12,[evalm(coeff(sim_dfdp_2_10k, P1,
1)),evalm(coeff(sim_dfdp_2_10k, P2, 1)),evalm(coeff(sim_dfdp_2_10k, P3,
1)),evalm(coeff(sim_dfdp_2_10k, P4, 1)),evalm(coeff(sim_dfdp_2_10k, P5,
1)),evalm(coeff(sim_dfdp_2_10k, P6, 1)),evalm(coeff(sim_dfdp_2_10k, P7,
1)),evalm(coeff(sim_dfdp_2_10k, P8, 1)),evalm(coeff(sim_dfdp_2_10k, P9,
1)),evalm(coeff(sim_dfdp_2_10k, P10, 1)),evalm(coeff(sim_dfdp_2_10k, P11,
1)),evalm(coeff(sim_dfdp_2_10k, P12, 1))]):
> tem:= evalm(B_2_10k.params):
> A_2_10k:=matrix(1,1,[evalm(sim_dfdp_2_10k-tem)]):
>
> d_phi_2_11k:=d_phi[2,11]:
> f_2_11k:=subs(Para_set, evalm(d_phi_2_11k)):
> dfdp_2_11k:=grad(d_phi_2_11k,params):
> dfdp_2_11k:=subs(Para_set,evalm(dfdp_2_11k)):
> sim_dfdp_2_11k:=evalm(evalm(dfdp_2_11k).P_sub)+f_2_11k:
> B_2_11k:=matrix(1,12,[evalm(coeff(sim_dfdp_2_11k, P1,
1)),evalm(coeff(sim_dfdp_2_11k, P2, 1)),evalm(coeff(sim_dfdp_2_11k, P3,
1)),evalm(coeff(sim_dfdp_2_11k, P4, 1)),evalm(coeff(sim_dfdp_2_11k, P5,
1)),evalm(coeff(sim_dfdp_2_11k, P6, 1)),evalm(coeff(sim_dfdp_2_11k, P7,
1)),evalm(coeff(sim_dfdp_2_11k, P8, 1)),evalm(coeff(sim_dfdp_2_11k, P9,
1)),evalm(coeff(sim_dfdp_2_11k, P10, 1)),evalm(coeff(sim_dfdp_2_11k, P11,
1)),evalm(coeff(sim_dfdp_2_11k, P12, 1))]):
> tem:= evalm(B_2_11k.params):
> A_2_11k:=matrix(1,1,[evalm(sim_dfdp_2_11k-tem)]):
>
> d_phi_2_12k:=d_phi[2,12]:
> f_2_12k:=subs(Para_set, evalm(d_phi_2_12k)):
> dfdp_2_12k:=grad(d_phi_2_12k,params):
> dfdp_2_12k:=subs(Para_set,evalm(dfdp_2_12k)):
> sim_dfdp_2_12k:=evalm(evalm(dfdp_2_12k).P_sub)+f_2_12k:
> B_2_12k:=matrix(1,12,[evalm(coeff(sim_dfdp_2_12k, P1,
1)),evalm(coeff(sim_dfdp_2_12k, P2, 1)),evalm(coeff(sim_dfdp_2_12k, P3,
1)),evalm(coeff(sim_dfdp_2_12k, P4, 1)),evalm(coeff(sim_dfdp_2_12k, P5,
1)),evalm(coeff(sim_dfdp_2_12k, P6, 1)),evalm(coeff(sim_dfdp_2_12k, P7,
1)),evalm(coeff(sim_dfdp_2_12k, P8, 1)),evalm(coeff(sim_dfdp_2_12k, P9,
1)),evalm(coeff(sim_dfdp_2_12k, P10, 1)),evalm(coeff(sim_dfdp_2_12k, P11,
1)),evalm(coeff(sim_dfdp_2_12k, P12, 1))]):
> tem:= evalm(B_2_12k.params):
> A_2_12k:=matrix(1,1,[evalm(sim_dfdp_2_12k-tem)]):
>
>
A2:=stackmatrix(A_2_1k,A_2_2k,A_2_3k,A_2_4k,A_2_5k,A_2_6k,A_2_7k,A_2_8k,A_2_9k,A_2
_10k,A_2_11k,A_2_12k):
>
B2:=stackmatrix(B_2_1k,B_2_2k,B_2_3k,B_2_4k,B_2_5k,B_2_6k,B_2_7k,B_2_8k,B_2_9k,B_2
_10k,B_2_11k,B_2_12k):
> P_nom:=matrix(12,1,Para_nom):
> As2:=evalm(evalm(A2/s)-P_nom):
> T2:=evalm((diag(s,s,s,s,s,s,s,s,s,s,s,s)-B2)):
> Tinv2:=evalm(inverse(T2)):
> PL2:=evalm(Tinv2&*As2):
> pT2:=map(invlaplace,evalm(PL2),s,t2):
>
>
> d_phi_3_1k:=d_phi[3,1]:
> f_3_1k:=subs(Para_set, evalm(d_phi_3_1k)):
> dfdp_3_1k:=grad(d_phi_3_1k,params):
> dfdp_3_1k:=subs(Para_set,evalm(dfdp_3_1k)):
> sim_dfdp_3_1k:=evalm(evalm(dfdp_3_1k).P_sub)+f_3_1k:
> B_3_1k:=matrix(1,12,[evalm(coeff(sim_dfdp_3_1k, P1,
```

```
1)),evalm(coeff(sim_dfdp_3_1k, P2, 1)),evalm(coeff(sim_dfdp_3_1k, P3,
1)),evalm(coeff(sim_dfdp_3_1k, P4, 1)),evalm(coeff(sim_dfdp_3_1k, P5,
1)),evalm(coeff(sim_dfdp_3_1k, P6, 1)),evalm(coeff(sim_dfdp_3_1k, P7,
1)),evalm(coeff(sim_dfdp_3_1k, P8, 1)),evalm(coeff(sim_dfdp_3_1k, P9,
1)),evalm(coeff(sim_dfdp_3_1k, P10, 1)),evalm(coeff(sim_dfdp_3_1k, P11,
1)),evalm(coeff(sim_dfdp_3_1k, P12, 1))]):
> tem:= evalm(B_3_1k.params):
> A_3_1k:=matrix(1,1,[evalm(sim_dfdp_3_1k-tem)]):
>
> d_phi_3_2k:=d_phi[3,2]:
> f_3_2k:=subs(Para_set, evalm(d_phi_3_2k)):
> dfdp_3_2k:=grad(d_phi_3_2k,params):
> dfdp_3_2k:=subs(Para_set,evalm(dfdp_3_2k)):
> sim_dfdp_3_2k:=evalm(evalm(dfdp_3_2k).P_sub)+f_3_2k:
> B_3_2k:=matrix(1,12,[evalm(coeff(sim_dfdp_3_2k, P1,
1)),evalm(coeff(sim_dfdp_3_2k, P2, 1)),evalm(coeff(sim_dfdp_3_2k, P3,
1)),evalm(coeff(sim_dfdp_3_2k, P4, 1)),evalm(coeff(sim_dfdp_3_2k, P5,
1)),evalm(coeff(sim_dfdp_3_2k, P6, 1)),evalm(coeff(sim_dfdp_3_2k, P7,
1)),evalm(coeff(sim_dfdp_3_2k, P8, 1)),evalm(coeff(sim_dfdp_3_2k, P9,
1)),evalm(coeff(sim_dfdp_3_2k, P10, 1)),evalm(coeff(sim_dfdp_3_2k, P11,
1)),evalm(coeff(sim_dfdp_3_2k, P12, 1))]):
> tem:= evalm(B_3_2k.params):
> A_3_2k:=matrix(1,1,[evalm(sim_dfdp_3_2k-tem)]):
>
> d_phi_3_3k:=d_phi[3,3]:
> f_3_3k:=subs(Para_set, evalm(d_phi_3_3k)):
> dfdp_3_3k:=grad(d_phi_3_3k,params):
> dfdp_3_3k:=subs(Para_set,evalm(dfdp_3_3k)):
> sim_dfdp_3_3k:=evalm(evalm(dfdp_3_3k).P_sub)+f_3_3k:
> B_3_3k:=matrix(1,12,[evalm(coeff(sim_dfdp_3_3k, P1,
1)),evalm(coeff(sim_dfdp_3_3k, P2, 1)),evalm(coeff(sim_dfdp_3_3k, P3,
1)),evalm(coeff(sim_dfdp_3_3k, P4, 1)),evalm(coeff(sim_dfdp_3_3k, P5,
1)),evalm(coeff(sim_dfdp_3_3k, P6, 1)),evalm(coeff(sim_dfdp_3_3k, P7,
1)),evalm(coeff(sim_dfdp_3_3k, P8, 1)),evalm(coeff(sim_dfdp_3_3k, P9,
1)),evalm(coeff(sim_dfdp_3_3k, P10, 1)),evalm(coeff(sim_dfdp_3_3k, P11,
1)),evalm(coeff(sim_dfdp_3_3k, P12, 1))]):
> tem:= evalm(B_3_3k.params):
> A_3_3k:=matrix(1,1,[evalm(sim_dfdp_3_3k-tem)]):
>
> d_phi_3_4k:=d_phi[3,4]:
> f_3_4k:=subs(Para_set, evalm(d_phi_3_4k)):
> dfdp_3_4k:=grad(d_phi_3_4k,params):
> dfdp_3_4k:=subs(Para_set,evalm(dfdp_3_4k)):
> sim_dfdp_3_4k:=evalm(evalm(dfdp_3_4k).P_sub)+f_3_4k:
> B_3_4k:=matrix(1,12,[evalm(coeff(sim_dfdp_3_4k, P1,
1)),evalm(coeff(sim_dfdp_3_4k, P2, 1)),evalm(coeff(sim_dfdp_3_4k, P3,
1)),evalm(coeff(sim_dfdp_3_4k, P4, 1)),evalm(coeff(sim_dfdp_3_4k, P5,
1)),evalm(coeff(sim_dfdp_3_4k, P6, 1)),evalm(coeff(sim_dfdp_3_4k, P7,
1)),evalm(coeff(sim_dfdp_3_4k, P8, 1)),evalm(coeff(sim_dfdp_3_4k, P9,
1)),evalm(coeff(sim_dfdp_3_4k, P10, 1)),evalm(coeff(sim_dfdp_3_4k, P11,
1)),evalm(coeff(sim_dfdp_3_4k, P12, 1))]):
> tem:= evalm(B_3_4k.params):
> A_3_4k:=matrix(1,1,[evalm(sim_dfdp_3_4k-tem)]):
>
> d_phi_3_5k:=d_phi[3,5]:
> f_3_5k:=subs(Para_set, evalm(d_phi_3_5k)):
> dfdp_3_5k:=grad(d_phi_3_5k,params):
> dfdp_3_5k:=subs(Para_set,evalm(dfdp_3_5k)):
> sim_dfdp_3_5k:=evalm(evalm(dfdp_3_5k).P_sub)+f_3_5k:
> B_3_5k:=matrix(1,12,[evalm(coeff(sim_dfdp_3_5k, P1,
1)),evalm(coeff(sim_dfdp_3_5k, P2, 1)),evalm(coeff(sim_dfdp_3_5k, P3,
1)),evalm(coeff(sim_dfdp_3_5k, P4, 1)),evalm(coeff(sim_dfdp_3_5k, P5,
1)),evalm(coeff(sim_dfdp_3_5k, P6, 1)),evalm(coeff(sim_dfdp_3_5k, P7,
1)),evalm(coeff(sim_dfdp_3_5k, P8, 1)),evalm(coeff(sim_dfdp_3_5k, P9,
1)),evalm(coeff(sim_dfdp_3_5k, P10, 1)),evalm(coeff(sim_dfdp_3_5k, P11,
1)),evalm(coeff(sim_dfdp_3_5k, P12, 1))]):
> tem:= evalm(B_3_5k.params):
> A_3_5k:=matrix(1,1,[evalm(sim_dfdp_3_5k-tem)]):
>
> d_phi_3_6k:=d_phi[3,6]:
> f_3_6k:=subs(Para_set, evalm(d_phi_3_6k)):
```

```
> dfdp_3_6k:=grad(d_phi_3_6k,params):
> dfdp_3_6k:=subs(Para_set,evalm(dfdp_3_6k)):
> sim_dfdp_3_6k:=evalm(evalm(dfdp_3_6k).P_sub)+f_3_6k:
> B_3_6k:=matrix(1,12,[evalm(coeff(sim_dfdp_3_6k, P1,
1)),evalm(coeff(sim_dfdp_3_6k, P2, 1)),evalm(coeff(sim_dfdp_3_6k, P3,
1)),evalm(coeff(sim_dfdp_3_6k, P4, 1)),evalm(coeff(sim_dfdp_3_6k, P5,
1)),evalm(coeff(sim_dfdp_3_6k, P6, 1)),evalm(coeff(sim_dfdp_3_6k, P7,
1)),evalm(coeff(sim_dfdp_3_6k, P8, 1)),evalm(coeff(sim_dfdp_3_6k, P9,
1)),evalm(coeff(sim_dfdp_3_6k, P10, 1)),evalm(coeff(sim_dfdp_3_6k, P11,
1)),evalm(coeff(sim_dfdp_3_6k, P12, 1))]):
> tem:= evalm(B_3_6k.params):
> A_3_6k:=matrix(1,1,[evalm(sim_dfdp_3_6k-tem)]):
>
> d_phi_3_7k:=d_phi[3,7]:
> f_3_7k:=subs(Para_set, evalm(d_phi_3_7k)):
> dfdp_3_7k:=grad(d_phi_3_7k,params):
> dfdp_3_7k:=subs(Para_set,evalm(dfdp_3_7k)):
> sim_dfdp_3_7k:=evalm(evalm(dfdp_3_7k).P_sub)+f_3_7k:
> B_3_7k:=matrix(1,12,[evalm(coeff(sim_dfdp_3_7k, P1,
1)),evalm(coeff(sim_dfdp_3_7k, P2, 1)),evalm(coeff(sim_dfdp_3_7k, P3,
1)),evalm(coeff(sim_dfdp_3_7k, P4, 1)),evalm(coeff(sim_dfdp_3_7k, P5,
1)),evalm(coeff(sim_dfdp_3_7k, P6, 1)),evalm(coeff(sim_dfdp_3_7k, P7,
1)),evalm(coeff(sim_dfdp_3_7k, P8, 1)),evalm(coeff(sim_dfdp_3_7k, P9,
1)),evalm(coeff(sim_dfdp_3_7k, P10, 1)),evalm(coeff(sim_dfdp_3_7k, P11,
1)),evalm(coeff(sim_dfdp_3_7k, P12, 1))]):
> tem:= evalm(B_3_7k.params):
> A_3_7k:=matrix(1,1,[evalm(sim_dfdp_3_7k-tem)]):
>
> d_phi_3_8k:=d_phi[3,8]:
> f_3_8k:=subs(Para_set, evalm(d_phi_3_8k)):
> dfdp_3_8k:=grad(d_phi_3_8k,params):
> dfdp_3_8k:=subs(Para_set,evalm(dfdp_3_8k)):
> sim_dfdp_3_8k:=evalm(evalm(dfdp_3_8k).P_sub)+f_3_8k:
> B_3_8k:=matrix(1,12,[evalm(coeff(sim_dfdp_3_8k, P1,
1)),evalm(coeff(sim_dfdp_3_8k, P2, 1)),evalm(coeff(sim_dfdp_3_8k, P3,
1)),evalm(coeff(sim_dfdp_3_8k, P4, 1)),evalm(coeff(sim_dfdp_3_8k, P5,
1)),evalm(coeff(sim_dfdp_3_8k, P6, 1)),evalm(coeff(sim_dfdp_3_8k, P7,
1)),evalm(coeff(sim_dfdp_3_8k, P8, 1)),evalm(coeff(sim_dfdp_3_8k, P9,
1)),evalm(coeff(sim_dfdp_3_8k, P10, 1)),evalm(coeff(sim_dfdp_3_8k, P11,
1)),evalm(coeff(sim_dfdp_3_8k, P12, 1))]):
> tem:= evalm(B_3_8k.params):
> A_3_8k:=matrix(1,1,[evalm(sim_dfdp_3_8k-tem)]):
>
> d_phi_3_9k:=d_phi[3,9]:
> f_3_9k:=subs(Para_set, evalm(d_phi_3_9k)):
> dfdp_3_9k:=grad(d_phi_3_9k,params):
> dfdp_3_9k:=subs(Para_set,evalm(dfdp_3_9k)):
> sim_dfdp_3_9k:=evalm(evalm(dfdp_3_9k).P_sub)+f_3_9k:
> B_3_9k:=matrix(1,12,[evalm(coeff(sim_dfdp_3_9k, P1,
1)),evalm(coeff(sim_dfdp_3_9k, P2, 1)),evalm(coeff(sim_dfdp_3_9k, P3,
1)),evalm(coeff(sim_dfdp_3_9k, P4, 1)),evalm(coeff(sim_dfdp_3_9k, P5,
1)),evalm(coeff(sim_dfdp_3_9k, P6, 1)),evalm(coeff(sim_dfdp_3_9k, P7,
1)),evalm(coeff(sim_dfdp_3_9k, P8, 1)),evalm(coeff(sim_dfdp_3_9k, P9,
1)),evalm(coeff(sim_dfdp_3_9k, P10, 1)),evalm(coeff(sim_dfdp_3_9k, P11,
1)),evalm(coeff(sim_dfdp_3_9k, P12, 1))]):
> tem:= evalm(B_3_9k.params):
> A_3_9k:=matrix(1,1,[evalm(sim_dfdp_3_9k-tem)]):
>
> d_phi_3_10k:=d_phi[3,10]:
> f_3_10k:=subs(Para_set, evalm(d_phi_3_10k)):
> dfdp_3_10k:=grad(d_phi_3_10k,params):
> dfdp_3_10k:=subs(Para_set,evalm(dfdp_3_10k)):
> sim_dfdp_3_10k:=evalm(evalm(dfdp_3_10k).P_sub)+f_3_10k:
> B_3_10k:=matrix(1,12,[evalm(coeff(sim_dfdp_3_10k, P1,
1)),evalm(coeff(sim_dfdp_3_10k, P2, 1)),evalm(coeff(sim_dfdp_3_10k, P3,
1)),evalm(coeff(sim_dfdp_3_10k, P4, 1)),evalm(coeff(sim_dfdp_3_10k, P5,
1)),evalm(coeff(sim_dfdp_3_10k, P6, 1)),evalm(coeff(sim_dfdp_3_10k, P7,
1)),evalm(coeff(sim_dfdp_3_10k, P8, 1)),evalm(coeff(sim_dfdp_3_10k, P9,
1)),evalm(coeff(sim_dfdp_3_10k, P10, 1)),evalm(coeff(sim_dfdp_3_10k, P11,
1)),evalm(coeff(sim_dfdp_3_10k, P12, 1))]):
> tem:= evalm(B_3_10k.params):
```

```
> A_3_10k:=matrix(1,1,[evalm(sim_dfdp_3_10k-tem)]):
>
> d_phi_3_11k:=d_phi[3,11]:
> f_3_11k:=subs(Para_set, evalm(d_phi_3_11k)):
> dfdp_3_11k:=grad(d_phi_3_11k,params):
> dfdp_3_11k:=subs(Para_set,evalm(dfdp_3_11k)):
> sim_dfdp_3_11k:=evalm(evalm(dfdp_3_11k).P_sub)+f_3_11k:
> B_3_11k:=matrix(1,12,[evalm(coeff(sim_dfdp_3_11k, P1,
1)),evalm(coeff(sim_dfdp_3_11k, P2, 1)),evalm(coeff(sim_dfdp_3_11k, P3,
1)),evalm(coeff(sim_dfdp_3_11k, P4, 1)),evalm(coeff(sim_dfdp_3_11k, P5,
1)),evalm(coeff(sim_dfdp_3_11k, P6, 1)),evalm(coeff(sim_dfdp_3_11k, P7,
1)),evalm(coeff(sim_dfdp_3_11k, P8, 1)),evalm(coeff(sim_dfdp_3_11k, P9,
1)),evalm(coeff(sim_dfdp_3_11k, P10, 1)),evalm(coeff(sim_dfdp_3_11k, P11,
1)),evalm(coeff(sim_dfdp_3_11k, P12, 1))]):
> tem:= evalm(B_3_11k.params):
> A_3_11k:=matrix(1,1,[evalm(sim_dfdp_3_11k-tem)]):
>
> d_phi_3_12k:=d_phi[3,12]:
> f_3_12k:=subs(Para_set, evalm(d_phi_3_12k)):
> dfdp_3_12k:=grad(d_phi_3_12k,params):
> dfdp_3_12k:=subs(Para_set,evalm(dfdp_3_12k)):
> sim_dfdp_3_12k:=evalm(evalm(dfdp_3_12k).P_sub)+f_3_12k:
> B_3_12k:=matrix(1,12,[evalm(coeff(sim_dfdp_3_12k, P1,
1)),evalm(coeff(sim_dfdp_3_12k, P2, 1)),evalm(coeff(sim_dfdp_3_12k, P3,
1)),evalm(coeff(sim_dfdp_3_12k, P4, 1)),evalm(coeff(sim_dfdp_3_12k, P5,
1)),evalm(coeff(sim_dfdp_3_12k, P6, 1)),evalm(coeff(sim_dfdp_3_12k, P7,
1)),evalm(coeff(sim_dfdp_3_12k, P8, 1)),evalm(coeff(sim_dfdp_3_12k, P9,
1)),evalm(coeff(sim_dfdp_3_12k, P10, 1)),evalm(coeff(sim_dfdp_3_12k, P11,
1)),evalm(coeff(sim_dfdp_3_12k, P12, 1))]):
> tem:= evalm(B_3_12k.params):
> A_3_12k:=matrix(1,1,[evalm(sim_dfdp_3_12k-tem)]):
>
>
A3:=stackmatrix(A_3_1k,A_3_2k,A_3_3k,A_3_4k,A_3_5k,A_3_6k,A_3_7k,A_3_8k,A_3_9k,A_3
_10k,A_3_11k,A_3_12k):
>
B3:=stackmatrix(B_3_1k,B_3_2k,B_3_3k,B_3_4k,B_3_5k,B_3_6k,B_3_7k,B_3_8k,B_3_9k,B_3
_10k,B_3_11k,B_3_12k):
> P_nom:=matrix(12,1,Para_nom):
> As3:=evalm(evalm(A3/s)-P_nom):
> T3:=evalm((diag(s,s,s,s,s,s,s,s,s,s,s,s)-B3)):
> Tinv3:=evalm(inverse(T3)):
> PL3:=evalm(Tinv3&*As3):
> pT3:=map(invlaplace,evalm(PL3),s,t3):
>
>


> pT4:=matrix(12,1,[t4,0,0,0,0,0,0,0,0,0,0,0]):
> pT5:=matrix(12,1,[0,t5,0,0,0,0,0,0,0,0,0,0]):
> pT6:=matrix(12,1,[0,0,0,t6,0,0,0,0,0,0,0,0]):
> pT7:=matrix(12,1,[0,0,0,0,t7,0,0,0,0,0,0,0]):
> pT8:=matrix(12,1,[0,0,0,0,0,0,t8,0,0,0,0,0]):
> pT9:=matrix(12,1,[0,0,0,0,0,0,0,t9,0,0,0,0]):
> pT10:=matrix(12,1,[0,0,0,0,0,0,0,0,t10,0,0,0]):
> pT11:=matrix(12,1,[0,0,0,0,0,0,0,0,0,t11,0]):
> pT12:=matrix(12,1,[0,0,0,0,0,0,0,0,0,0,0,t12]):
>
> F:=evalm(pT1+pT2+pT3+pT4+pT5+pT6+pT7+pT8+pT9+pT10+pT11+pT12):
>
> tim := [t1, t2, t3, t4, t5, t6, t7, t8, t9, t10, t11, t12]:
> tim_set:={t1=1, t2=1, t3=1, t4=1, t5=1, t6=1, t7=1, t8=1, t9=1, t10=1, t11=1,
t12=1}:
> d_F_1:=grad(F[1,1],tim):
> d_F_2:=grad(F[2,1],tim):
> d_F_3:=grad(F[3,1],tim):
> d_F_4:=grad(F[4,1],tim):
> d_F_5:=grad(F[5,1],tim):
> d_F_6:=grad(F[6,1],tim):
> d_F_7:=grad(F[7,1],tim):
> d_F_8:=grad(F[8,1],tim):
> d_F_9:=grad(F[9,1],tim):
```

```
> d_F_10:=grad(F[10,1],tim):
> d_F_11:=grad(F[11,1],tim):
> d_F_12:=grad(F[12,1],tim):
> DF1
:=stackmatrix(d_F_1,d_F_2,d_F_3,d_F_4,d_F_5,d_F_6,d_F_7,d_F_8,d_F_9,d_F_10,d_F_11,
d_F_12):
> DF:=evalf(subs(tim_set,evalm(DF1))):
>
> with(CodeGeneration):
> Matlab(F):
```

B

Experimental data

This appendix contains the experimental data from the implementation of the D-optimal input profile on a bioreactor system.

Table B.1 Measured data

Time (hours from start)	Measurable biomass concentration (X)	External nitrogen source concentration (S_1)	External carbon source concentration (S_2)	Mass fraction of algal oil stored in cells $\left(\frac{I_p}{X}\right)$
	(g/mL)	(g/mL)	(g/mL)	
0	0.000857	0.00045	41.4914	0.580
4	0.000857	0.000416	0.040844	0.509
8	0.000786	0.000404	0.040857	0.583
12	0.000857	0.000414	0.040703	0.597
16	0.00075	0.000416	0.040496	0.508
20	0.000857	0.000409	0.04077	0.456
24	0.001143	0.000365	0.040754	0.392
28	0.001036	0.000316	0.040704	0.444
32	0.001143	0.000251	0.040558	0.270
36	0.001071	0.000182	0.040125	0.355
40	0.001607	6.52E-05	0.039878	0.224
44	0.00175	6.26E-05	0.03969	0.180
48	0.002036	5.09E-05	0.039126	0.143
52	0.002286	5.43E-05	0.038423	0.150
56	0.002429	5.48E-05	0.037285	0.156
68	0.004821	5.89E-05	0.032107	0.216
72	0.005179	7.21E-05	0.045365	0.235
76	0.005607	6.45E-05	0.06625	0.230
80	0.005429	7.41E-05	0.081869	0.252
84	0.005525	8.22E-05	0.091911	0.282
88	0.0055	8.9E-05	0.096808	0.274
92	0.005679	8.49E-05	0.100676	0.266
96	0.005857	8.41E-05	0.101893	0.341
100	0.006179	5.27E-05	0.101557	0.399
104	0.006494	5.18E-05	0.100441	0.408
108	0.006673	5.24E-05	0.09891	0.392
112	0.007179	4.35E-05	0.098409	0.385
116	0.007643	7.18E-05	0.097243	0.454

120	0.007964	8.56E-05	0.096991	0.486
124	0.008643	7.46E-05	0.095054	0.477
128	0.00858	8.13E-05	0.092885	0.389
132	0.009179	5.42E-05	0.091017	0.399
136	0.009893	5.49E-05	0.089653	0.413
140	0.010571	4.53E-05	0.087814	0.350
148	0.013714	6.04E-05	0.087706	0.391
164	0.012893	0.000105	0.07983	0.410
168	0.013286	7.03E-05	0.076217	0.437
176	0.014464	7.35E-05	0.073618	0.473
180	0.015286	7.29E-05	0.071009	0.426
184	0.016419	7.92E-05	0.06904	0.433
188	0.017107	7.08E-05	0.067683	0.470
192	0.017357	7.69E-05	0.06629	0.473
196	0.018036	7.97E-05	0.064663	0.554
200	0.018304	7.78E-05	0.06143	0.477
204	0.018967	7.88E-05	0.060409	0.513
208	0.018643	7.97E-05	0.058652	0.432
212	0.021464	8.21E-05	0.056796	0.412
216	0.022321	8.42E-05	0.054916	0.372
220	0.022179	8.33E-05	0.053182	0.432
224	0.022974	8.2E-05	0.052098	0.432
228	0.023415	8.1E-05	0.050573	0.498
232	0.023357	7.8E-05	0.048532	0.472
236	0.025607	8.57E-05	0.046809	0.492
240	0.025393	7.68E-05	0.044029	0.451
244	0.025536	9.37E-05	0.040607	0.512
248	0.029512	8.97E-05	0.040998	0.524
260	0.02875	8.65E-05	0.034833	0.508
264	0.029929	8.2E-05	0.032701	0.460
268	0.030143	9.39E-05	0.031534	0.463
272	0.031565	0.000102	0.029629	0.441
276	0.0325	9.45E-05	0.028113	0.417
280	0.032333	9.02E-05	0.026599	0.508
284	0.034821	9.13E-05	0.024665	0.575
288	0.034679	6.41E-05	0.022899	0.545
292	0.035846	8.45E-05	0.021662	0.610
308	0.035821	0.000103	0.015355	0.562
312	0.036143	0.0001	0.013536	0.608
316	0.033893	9.61E-05	0.013138	0.561
320	0.03574	9.69E-05	0.012461	0.543
324	0.035385	9.82E-05	0.011779	0.558
328	0.036571	9.46E-05	0.009776	0.532
332	0.03675	9.32E-05	0.007653	0.583
336	0.03725	0.000108	0.005553	0.582
340	0.039143	9.9E-05	0.003387	0.615
344	0.03875	9.7E-05	0.001139	0.651
348	0.039357	8.69E-05	1.67E-05	0.551
352	0.039	9E-05	1.29E-05	0.583
356	0.039	9.22E-05	5.74E-06	0.632
360	0.039214	8.74E-05	1.55E-05	0.627