University of Kentucky
UKnowledge

Theses and Dissertations--Chemical and Materials Engineering

Chemical and Materials Engineering

2018

APPLICATION OF PROCESS SYSTEMS ENGINEERING TOOLS AND METHODS TO FERMENTATION-BASED BIOREFINERIES

Kwabena Darkwah
University of Kentucky, mydarkoline@yahoo.com
Author ORCID Identifier:
https://orcid.org/0000-0003-4283-6052

Digital Object Identifier: https://doi.org/10.13023/ETD.2018.035

Right click to open a feedback form in a new tab to let us know how this document benefits you.

Recommended Citation

Darkwah, Kwabena, "APPLICATION OF PROCESS SYSTEMS ENGINEERING TOOLS AND METHODS TO FERMENTATION-BASED BIOREFINERIES" (2018). *Theses and Dissertations--Chemical and Materials Engineering*. 83.
https://uknowledge.uky.edu/cme_etds/83

This Doctoral Dissertation is brought to you for free and open access by the Chemical and Materials Engineering at UKnowledge. It has been accepted for inclusion in Theses and Dissertations--Chemical and Materials Engineering by an authorized administrator of UKnowledge. For more information, please contact UKnowledge@lsv.uky.edu.

STUDENT AGREEMENT:

I represent that my thesis or dissertation and abstract are my original work. Proper attribution has been given to all outside sources. I understand that I am solely responsible for obtaining any needed copyright permissions. I have obtained needed written permission statement(s) from the owner(s) of each third-party copyrighted matter to be included in my work, allowing electronic distribution (if such use is not permitted by the fair use doctrine) which will be submitted to UKnowledge as Additional File.

I hereby grant to The University of Kentucky and its agents the irrevocable, non-exclusive, and royalty-free license to archive and make accessible my work in whole or in part in all forms of media, now or hereafter known. I agree that the document mentioned above may be made available immediately for worldwide access unless an embargo applies.

I retain all other ownership rights to the copyright of my work. I also retain the right to use in future works (such as articles or books) all or part of my work. I understand that I am free to register the copyright to my work.

REVIEW, APPROVAL AND ACCEPTANCE

The document mentioned above has been reviewed and accepted by the student’s advisor, on behalf of the advisory committee, and by the Director of Graduate Studies (DGS), on behalf of the program; we verify that this is the final, approved version of the student’s thesis including all changes required by the advisory committee. The undersigned agree to abide by the statements above.

Kwabena Darkwah, Student

Dr. Barbara L. Knutson, Major Professor

Dr. Thomas D. Dziubla, Director of Graduate Studies

APPLICATION OF PROCESS SYSTEMS ENGINEERING TOOLS AND METHODS TO FERMENTATION-BASED BIOREFINERIES

DISSERTATION

A dissertation submitted in partial fulfillment of the
requirements for the degree of Doctor of Philosophy in the
College of Engineering
at the University of Kentucky

By
Kwabena Darkwah

Lexington, Kentucky

Co-Directors: Dr. Barbara L. Knutson, Professor of Chemical Engineering
and Dr. Jeffrey Seay, Associate Professor of Chemical Engineering

Lexington, Kentucky
2018

Copyright© Kwabena Darkwah 2018

ABSTRACT OF DISSERTATION

APPLICATION OF PROCESS SYSTEMS ENGINEERING TOOLS AND METHODS TO FERMENTATION-BASED BIOREFINERIES

Biofuels produced from lignocellulosic biomass via the fermentation platform are sustainable energy alternatives to fossil fuels. Process Systems Engineering (PSE) uses computer-based tools and methods to design, simulate and optimize processes. Application of PSE tools to the design of economic biorefinery processes requires the development of simulation approaches that can be integrated with existing, mature PSE tools used to optimize traditional refineries, such as Aspen Plus. Current unit operation models lack the ability to describe unsteady state fermentation processes, link unsteady state fermentation with *in situ* separations, and optimize these processes for competing factors (e.g., yield and productivity). This work applies a novel architecture of commercial PSE tools, Aspen Plus and MATLAB, to develop techniques to simulate time-dependent fermentation without and with *in situ* separations for process design, analyses and optimization of the operating conditions.

Traditional batch fermentation simulations with *in situ* separations decouple these interdependent steps in a separate “steady state” reactor followed by an equilibrium separation of the final fermentation broth. A typical mechanistic system of ordinary differential equations (ODEs) describing a batch fermentation does not fit the standard built-in power law reaction kinetics model in Aspen Plus. To circumvent this challenge, a novel platform that links the batch reactor to a FORTRAN user kinetics subroutine (incorporates the ODEs) combined with component substitution (to simulate non-databank components) is utilized to simulate an unsteady state batch and *in situ* gas stripping process. The resulting model system predicts the product profile to be sensitive to the gas flow rate unlike previous “steady state” simulations. This demonstrates the importance of linking a time-dependent fermentation model to the fermentation environment for the design and analyses of fermentation processes.

A novel platform linking the genetic algorithm multi-objective and single-objective optimizations in MATLAB to the unsteady state batch fermentation simulation in Aspen

Plus through a component object module communication platform is utilized to optimize the operating conditions of a typical batch fermentation process. Two major contributions are: prior concentration of sugars from a typical lignocellulosic hydrolysate may be needed and with a higher initial sugar concentration, the fermentation process must be integrated with an *in situ* separation process to optimize the performance of fermentation processes. With this framework, fermentation experimentalists can use the full suite of PSE tools and methods to integrate biorefineries and refineries and as a decision-support tool to guide the design, analyses and optimization of fermentation-based biorefineries.

KEYWORDS: Lignocellulosic biomass, Aspen Plus unsteady state simulation, FORTRAN user kinetics subroutine, multi-objective optimization, sugar platform

Kwabena Darkwah

February 9, 2018

APPLICATION OF PROCESS SYSTEMS ENGINEERING TOOLS AND METHODS TO FERMENTATION-BASED BIOREFINERIES

By

Kwabena Darkwah

Dr. Barbara L. Knutson
Co-Director of Dissertation

Dr. Jeffrey Seay
Co-Director of Dissertation

Dr. Thomas D. Dziubla
Director of Graduate Studies

February 9, 2018

ACKNOWLEDGEMENTS

The successful completion of this work has been a collective effort of so many people. My deepest gratitude goes to my advisor, Dr. Barbara Knutson, for her unwavering support in and out of school, the professional training I received under her guidance, insightful critiques and pushing me to be the researcher that I am today. I thank my co-advisor, Dr. Jeffrey Seay, for his availability, dependable support in many diverse ways, the professional network he introduced me to and his expertise in guiding my work. Dr. Seay, you rock! Dr. Stephen Rankin contributed to my work and well-being in many diverse ways, and for that, I am very grateful. I would also like to thank Dr. Sues Nokes and Dr. Fazleena Badurdeen for being great collaborators and enthusiastically contributing to my work and professional development. To all my professors and all other professors in the department, thank you for your dedication to graduate education.

I am also very thankful to my lab mates and research associates, who in many diverse ways provided the much-needed support. I thank Dr. Shanshan Zhou, Dr. Daniel Schlipf, Dr. Sumesh Sukumara, Dr. Syed Islam, Arif Khan, Yuxin He, Masha Moradipour for their engaging discussions and support. To my friend, Melissia Witt, thank you so much for providing a shoulder to cry on and listening to my frustrations when things were tough.

Finally, I thank Ama Ohene-Addo, my brother, George Obiri Larbi, and my family for keeping me grounded. Thank you to my friend, Chandi Joshi, for being an incredible and dependable friend, and shouting “Aluta Contiua!” with me. To dearest wife and the love of my life, I say thank you for your unconditional love and support. We made it, honey.

Thanks be to the God of Abraham, Isaac and Jacob!

TABLE OF CONTENTS

ACKNOWLEDGEMENTS iii

LIST OF TABLES x

LIST OF FIGURES xi

Chapter 1 1

Introduction 1

1.1 Introduction 1
1.2 Research objectives 4

Chapter 2 8

Background 8

2.1 Bio-based products as sustainable alternatives to fossil-based products 8
2.2 Biorefinery 9
  2.2.1 Biorefinery definition 9
  2.2.2 Biorefinery classification 9
  2.2.3 Similarities and differences between a biorefinery and refinery 9
2.3 Classification of the feed stock for biorefineries 12
  2.3.1 First generation biomass 12
  2.3.2 Second generation biomass 12
  2.3.3 Third generation biomass 12
  2.3.4 Fourth generation biomass 12
2.4 The process of converting lignocellulosic biomass into bioproducts via fermentation 12
  2.4.1 Pretreatment 13
  2.4.2 Hydrolysis 13
  2.4.3 Fermentation 14
2.5 Fermentation design strategies 15
  2.5.1 Gas stripping 17
  2.5.2 Adsorption 18
  2.5.3 Pervaporation 19

2.6 Application of Process Systems Engineering to fermentation-based biorefineries 19

Chapter 3 25

A Perspective on Challenges and Prospects for Applying Process Systems Engineering Tools to Fermentation-Based Biorefineries 25

3.1 Summary 25
3.2 Introduction 25
3.3 Background 27
  3.3.1 Biorefining 27
  3.3.2 Process Systems Engineering 29
3.4 The role and challenges of PSE in process design, simulation, and optimization .. 29
  3.4.1 Challenges in using commercial and mature PSE tools for fermentation-based simulations 30
    3.4.1.1 Databases and physical property methods for bio-based components in traditional PSE tools 31
  3.4.2 Challenges in applying traditional PSE process simulators for optimization of fermentation-based biorefineries. 32
    3.4.2.1 Examples of using traditional process simulators for optimization of fermentation-based biorefineries. 35
  3.4.3 Incorporation of rate kinetics in fermentation simulations using traditional process simulators. 36
  3.4.4 Traditional simulation approaches for fermentation process in biorefineries .. 37
  3.4.5 Current state of PSE for fermentation process modeling 38
    3.4.5.1 The gap between developments in biology and mathematical models. 40
3.5 A proposed integrated platform of PSE tools for time-dependent fermentation processes 41
3.6 Conclusions and future directions 44

Chapter 4 47

Unsteady State Process Simulation of Integrated Batch Acetone-Butanol-Ethanol (ABE) Fermentation and in situ Gas Stripping 47

4.1 Summary 47
4.2 Introduction 47
4.3 Methods 50
  4.3.1 Prerequisites for batch fermentation simulation in Aspen Plus 50
    4.3.1.1 Fermentation model used in the simulation 50
    4.3.1.2 Non-databank components in the simulation 51
    4.3.1.3 Thermodynamic models in the simulation 51
  4.3.2 Simulation procedure for batch fermentation using Aspen Plus 52
    4.3.2.1 Interfacing the continuous steady state flowsheet environment in Aspen Plus with the unsteady state batch reactor 53
    4.3.2.2 Incorporating biological ODEs into the kinetics of the batch reactor in Aspen Plus 55
    4.3.2.3 Communication between the batch reactor and user kinetics subroutine and running the RBatch in Aspen Plus 56
    4.3.2.4 MATLAB simulations to verify unsteady state Aspen Plus ABE batch fermentation results 57
  4.3.3 Unsteady state fermentation and in situ gas stripping simulations 58
  4.3.4 Simulation of separate steady state fermentation and gas stripping processes. 58
  4.3.5 Calculation of ABE fermentation performance parameters 59
4.4 Results and discussion 59
  4.4.1 Validation of the Aspen Plus unsteady state batch fermentation with MATLAB simulation results 59
  4.4.2 Effect of gas flow rate on the fermentor and condensate ABE concentrations, total ABE produced, productivity and yield 61
  4.4.3 Comparison of performance of batch ABE fermentation and in situ gas stripping simulations to available literature. 63
  4.4.4 Comparison of traditional Aspen Plus batch separate steady state fermentation and gas stripping with unsteady state batch fermentation and in situ gas stripping simulations. 65
4.5 Conclusion 67

Chapter 5 69

Case study: Application of PSE Tools to the Design of a Fermentation Reactor with in situ Gas Stripping 69
5.1 Introduction 69
5.2 Process optimization 70
5.2.1 Fermentation model used in the simulation.. 72
5.2.2 Batch fermentation simulation in Aspen Plus 72
5.3 Results and discussion... 73
5.4 Conclusion... 76
Chapter 6 77
Multi-objective versus Single-objective Optimization of the Batch Bioethanol Production Based on a Time-dependent Fermentation Model.................................. 77
6.1 Summary 77
6.2 Introduction 77
6.3 Methods... 80
6.3.1 Integrated MATLAB-Aspen Plus platform for optimization of batch ethanolic process based on time-dependent fermentation model... 80
6.3.2 Prerequisites for process optimization of the batch ethanolic fermentation..... 81
6.3.2.1 Fermentation model used to simulate batch ethanolic fermentation 81
6.3.3 Linking MATLAB and Aspen Plus for a two-way communication 83
6.3.4 Batch ethanolic fermentation process optimization... 83
6.3.4.1 Initialization of optimization and generation of initial population (Pk) 85
6.3.4.2 Determination of objective function values.. 85
6.3.4.3 Optimization in MATLAB 86
6.3.4.4 Analyses of optimization results: MOO versus SOO 87
6.3.5 Optimization of integrated batch ethanolic fermentation and in situ product recovery 88
6.4 Results and Discussion... 89
6.4.1 Optimization of batch ethanolic fermentation using MOO and SOO 89
6.4.2 Optimization of integrated batch ethanolic fermentation with in situ separation using MOO and SOO 93

6.5 Conclusion.. 98

Chapter 7 99

Conclusion and Future Directions.. 99

7.1 Conclusion.. 99

7.2 Future work and directions.. 101

Appendices.. 105

Appendix A: Ordinary differential equations representation of the fermentation kinetics of a batch culture of Clostridium acetobutylicum.[120] 105

Appendix B: Parameter definition for the kinetic model and their respective values.[120] 106

Appendix C: Description of the FORTRAN user kinetic subroutine and running the Aspen Plus RBatch block and FOTRAN code.. 107

C.1 Description of the FORTRAN user kinetic subroutine and running the Aspen Plus RBatch block 107

C.2: FOTRAN code incorporating user kinetics subroutine 108

Appendix D: Procedure for the compilation of the Fortran user kinetic subroutine code to a readable Aspen Plus format.. 113

Appendix E: Stoichiometric equations and coefficients used in simulating a steady state batch fermentation in Aspen Plus, using a stoichiometric reactor 114

Appendix F: Graphs comparing simulation trends in Aspen Plus RBatch with the integration of ODEs describing the batch fermentation process in MATLAB........... 115

Appendix G: Comparison of simulation trends using the traditional Aspen Plus steady state approach that simulate the fermentation process entirely separate from the gas stripping process with the integrated batch ABE and in situ gas stripping simulated with the RBatch block linked to the Fortran user kinetic subroutine................................. 117

Appendix H: Comparison of the selectivity and percent recovery of acetone, butanol, ethanol, acetic acid, butyric acid and water at different gas flow rates per L of the fermentation broth for a batch fermentation and in situ gas stripping process. 118

Appendix I: Concentration of ABE, acetic and butyric acids in the condensate (stripped stream) and reactor using different gas flow rates per L of the fermentation broth for a batch fermentation and in situ gas stripping process 119

Appendix J: Formulation of the MOO and SOO problems used in case study in Chapter 5. 120

Appendix K: Details of the Net Flow Method (NFM)[138] and MATLAB sample code for scoring a set of Pareto optimal points using the NFM. 121

K.1: Details of the Net Flow Method (NFM).[138] 121

K.2: MATLAB sample code for scoring a set of Pareto optimal points using the NFM. 124

Appendix L: ODEs representation of the batch ethanolic fermentation kinetics of a batch culture of Z. mobilis and parameter definition for the kinetic model and their respective values.[137] 128

Appendix M: Creating a two-way communication between Aspen Plus and MATLAB to write and read data between the two platforms 131

Appendix N: Preliminary Aspen Plus results of an unsteady state fed-batch without and with in situ separations (gas stripping). 132

Appendix O: MOO and SOO code for the batch ABE fermentation with in situ gas stripping used in the Chapter 5 for the case study 136

O.1: Main MOO code 136

O.2: Output function for MOO code (visualize Pareto front and population changes) 139

O.3: SOO MATLAB code 140

Appendix P: MOO for the batch ethanolic fermentation with in situ gas stripping used in the Chapter 6 144

P.1: Main MOO code 144

P.2: Output function for MOO code 147

References 149

Vita 158

LIST OF TABLES

Table 2.1 Comparison of refineries and biorefineries based on the feedstock, building block chemical composition, processing and building block intermediates produced on the commercial scale.[44] 11

Table 2.2 Using Aspen Plus as a model process simulator to highlight the challenges for applying commercial process simulators to fermentation processes. 22

Table 3.1 Assessment of the databank and nondatabank components, and built-in rate based reaction kinetics and optimization tools available in Aspen Plus V9 and SuperPro Designer® v9.0. 34

Table 6.1 Comparison of the predicted decision variables and corresponding objective function values of the best optimal solution (ranked with NFM) in the MOO for the batch ethanolic fermentation without and with in situ separations 96

LIST OF FIGURES

Figure 2.1 Flowchart showing the classification of biorefineries based on the maturity of the technology process, conversion process, type of intermediate product, and type of feed stock used.[44] 10

Figure 2.2 The similarities between the concept of refineries and biorefineries.[45] 10

Figure 2.3 Schematic of the pretreatment and hydrolysis of lignocellulosic biomass to produce fermentable sugars.[53] 14

Figure 2.4 Schematic of the metabolism of biomass by solventogenic Clostridia.[5, 61] 16

Figure 2.5 Schematic of different fermentation schemes 17

Figure 2.6 The process of translating a real process into a mathematical model using Process Systems Engineering tools and methods 20

Figure 2.7 Comparison of a refinery with a biorefinery to highlight the challenges for applying commercial process simulators to biorefinery processes. 21

Figure 2.8 General framework for optimization of fermentation process based on unsteady state simulations in commercial process simulators coupled with rigorous optimization.23

Figure 2.9 Single-objective versus multi-objective optimization approaches. 24

Figure 3.1 Flowchart showing the diversity of feedstock and the potential components involved in biorefineries based on the sugar or fermentation platform.[13-14] 28

Figure 3.2 Mechanistic models classification based on cell population.[93] 39

Figure 3.3 General framework for process simulation and optimization of fermentation processes based on time-dependent (kinetic) model. 43

Figure 4.1 Simulation of integrated batch ABE fermentation and *in situ* gas stripping ... 49

Figure 4.2 Flow chart showing the interface between the Aspen Plus steady state environment and interaction between the unsteady state batch reactor linked to Fortran user subroutine.. 54

Figure 4.3 The configuration of the RBatch block in Aspen Plus. Adapted and modified from Aspen Plus V8.8 Help.[122] 55

Figure 4.4 Comparison of batch fermentation simulation results in Aspen Plus (RBatch) with the integration of the ordinary differential equations describing the batch fermentation process in MATLAB for cells (A), glucose (B), acetone (C) and butanol (D)................ 60

Figure 4.5 Comparison of the concentrations of acetone (A), butanol (B) and ethanol (E) in the fermentor (J) and condensate (K) using different gas flow rates........................... 61

Figure 4.6 Total ABE produced (total concentration), productivity and yield from the simulated batch ABE fermentation and *in situ* gas stripping with different gas flow rates 63

Figure 4.7 ABE yield (A), ABE productivity (B), total ABE produced (C) and ABE Selectivity (D) versus normalized gas flow rates (L/min per L of fermentation broth) from batch fermentation and *in situ* gas stripping literature data from refs *b,*[124] *c,*[125] *d,*[123] *e,*[126] *a,*[42] *f,*[127] *g,*[128] *h,*[129] *i,*[130]Aspen Plus unsteady state batch fermentation and *in situ* gas stripping simulation (*TS*) and performance benchmark (- - - - - -) for a typical batch fermentation of 0.35 g/g ABE yield, 0.30 g/L/h ABE productivity and 20 g/L total ABE produced (chosen based on data from Qureshi and Blaschek)[11]...................................... 64

Figure 4.8 Selectivities and condensate concentrations at the end of 32-h batch fermentation and *in situ* gas stripping of acetone (A), butanol (B) and ethanol (E) using traditional steady state (stoichiometric reactor and flash unit) and unsteady state (RBatch) in Aspen Plus with different gas flow rates. 66

Figure 5.1 Application of the proposed framework to process simulation and optimization of an integrated batch and *in situ* gas stripping.. 70

Figure 5.2 Plot of the MOO results for the integrated batch and *in situ* gas stripping process A) Final population B) Decision variables C) ABE selectivity versus ABE concentration in the condensate D) Total ABE produced versus ABE concentration in the condensate E) Total ABE produced versus ABE selectivity. The red circle, black diamonds, grey squares and green asterisk represents the best (top ranked), first 5 %, next 45 % and the last 50 %, respectively, ranked using the NFM method. 74

Figure 5.3 Comparison of literature data taken refs *a*,[42] *b*,[124] *c*,[125] *d*,[123] *e*,[126] *f*,[127] *g*,[128] *h*,[129] *i*.[130] with the MOO results from the case study for a batch fermentation and *in situ* gas stripping process. 75

Figure 6.1 Framework of integrated MATLAB-Aspen Plus platform for optimization of batch ethanolic process based on time-dependent fermentation model. 83

Figure 6.2 Plot of the MOO results for batch ethanolic fermentation 90

Figure 6.3 Concentration profile of the contents of the fermentor for the best optimal solution in the MOO (ranked with NFM) of the batch ethanolic fermentation (blue line, ▬) and integrated batch ethanolic fermentation with *in situ* separations (red filled circle, ●) The notation is a) cells concentration b) glucose concentration c) xylose concentration d) ethanol concentration 93

Figure 6.4 Plot of the decision variables (ranked with NFM) in the MOO results for integrated batch ethanolic fermentation and *in situ* separation.. 94

Figure 6.5 Plot of the decision variables (ranked with NFM) in the MOO results for integrated batch ethanolic fermentation and *in situ* separation. 97

Chapter 1

Introduction

1.1 Introduction

Biofuels, such as bioethanol and biobutanol, produced from lignocellulosic biomass via fermentation are sustainable energy alternatives to fossil fuels.[1] Microorganisms are used to convert substrates (sugars) into bioproducts by fermentation.[2] The batch ethanolic and acetone-butanol-ethanol (ABE) fermentations are common fermentation processes used to produce bioethanol, and bioethanol and biobutanol, respectively. Microorganisms commonly used for the batch ethanolic fermentation on the sugar platform include *Saccharomyces cerevisiae* (yeast), *Zymomonas mobilis*, *Escherichia coli*, *Clostridium thermocellum* as well as filamentous fungi such as *Aspergillus sp*. etc.[3] For the ABE fermentation, solventogenic *Clostridia* species (such as *C. acetobutylicum, C. beijerinckii, C. saccharobutylicum* and *saccharoperbutylacetonicum* [4-5]) are used in fermentation. Microorganisms used in fermentation are susceptible to both (high) substrate and product inhibitions. For example, greater than 5% ethanol concentrations inhibits yeast in ethanol fermentation from glucose[6-7] with complete microbial growth inhibition above 40% (w/v) glucose.[8] Further, greater than 86 g/L and 127 g/L of ethanol completely ceased cell growth and ethanol production in *Z. mobilis* ZM4, respectively.[9] For the batch ABE fermentation, more than 100 g/L glucose inhibited ABE production by *C. beijerinckii* in a batch fermentation.[10]

As result of substrate and product inhibitions, dilute aqueous fermentation broths with low titers, yields, productivity, and total solvents are produced in batch fermentations. For instance, low final ABE concentrations (< 20 g/L ABE), low reactor productivities (< 0.3 g/L/h) and low ABE yield (0.28 – 0.33 g/g) are obtained in ABE fermentations.[11] Furthermore, about 4% wt. ethanol is produced from lignocellulosic biomass in fermentations versus about 17% wt. for first generation biomass-based ethanolic fermentaitons.[12] The energy required to concentrate and purify the resulting dilute aqueous fermentation broth is significantly higher at alcohol concentrations less than 4% wt.[7] To reduce the consequential downstream separation cost, the fermentation titers, yields and productivities can be improved. Various fermentation schemes such batch, fed-batch and continuous fermentation are integrated with *in situ* product recovery techniques such as gas stripping, pervaporation, adsorption, etc. to alleviate substrate and product inhibitions as discussed in Chapter 2.

The challenges in fermentation-based biorefineries include - the requirement of microorganisms that can selectively and simultaneously utilize, at least, the major sugars from the mixture of sugars produced in the lignocellulose hydrolysate (hexose: glucose, mannose, galactose, fructose; pentose: xylose, arabinose; others: lactose, sucrose).[13-14] It was shown that if both glucose and xylose can be used in ethanolic fermentations, the production cost can be reduced from \$1.65 to \$1.23.[15] Second, the typical concentrations of sugars from lignocellulosic hydrolysate may not be optimum for the production of optimum product concentrations to minimize the downstream separation cost. For example, 1 – 23 g/L glucose was produced from 2 – 6% (w/v) sulfuric acid pretreated sugar cane bagasse[16], 1 – 20 g/L glucose from ionic liquid pretreated cellulose[17-18]) and about 20 g/L xylose and 58 g/L glucose produced from hydrogen peroxide-acetic acid pretreated Jerusalem artichoke hydrolyzed with enzymes (RUT-C30, pectinase and xylanase).[19] Additionally, the fermentation environment (substrate limitation and inhibition, product inhibition and potential recycle streams) affect the performance of the microorganisms. Lastly, the processing time raises questions as to the ideal length of time to run the batch fermentation process or when *in situ* product recovery, used to alleviate product inhibition, should be started. These are challenges that can potentially be addressed by Process Systems Engineering (PSE), the use of model-based methods and tools for the design, analysis, optimization, operation and control of complex chemical, biological or physical processes.[20-21]

A biorefinery uses biomass to produce fuels, power and chemicals in a facility that combines biomass conversion technologies and equipment.[22-26] Unlike the traditional petrochemical, refinery and chemical industries, PSE tools and methods are not readily applicable to fermentation-based biorefineries for process design, analyses and optimization. The inherent unsteady state nature of fermentation processes, the difficulty in coupling the kinetics of raw material conversion (unsteady state pretreatments and hydrolysis of lignocellulosic biomass and fermentation) into bioproducts with the autocatalytic microbial growth and death are critical challenges hindering the application of PSE tools and methods for fermentation-based biorefineries. In contrast, refinery and chemical processes are routinely modeled and simulated in traditional process simulators for analysis and optimization.

An optimization process uses computer-based algorithms to find the optimal decision variables that maximizes, minimizes or find a target value of one or more objectives. For the optimization of fermentation-based biorefineries, the large number of process variables and conditions involved makes the process complex. Additionally, many of the desired performance criteria are inherently partially or fully competing in nature and would require simultaneous optimization of multiple objectives, subject to constraints. Examples of competing pairs of performance criteria include product quality and recovery cost, product selectivity and conversion, etc. Multi-objective optimization (MOO), the simultaneous optimization of multiple objectives, is an alternative to approach to the traditional single-objective optimization (SOO, weighted sum of multiple objectives into a single objective or using one objective as the main objective of a multiple objective system and transforming other objectives into additional constraints). MOO gives a global perspective of an optimization process, where multiple equally optimal solutions are obtained (local and global). In contrast to SOO that finds one solution, MOO allows the trade-off in the, often, competing objectives to be observed to support the understanding of process data and serve as a decision-support tool and guide.[27-28]

For the optimization of fermentation processes to be realistic, time-dependent models and rigorous thermodynamic models are required to evaluate the objectives and (sometimes) constraints. Most commercial PSE tools lack in-built unit operation models with features that can readily simulate fermentation processes (e.g. the simulation of batch fermentation as unsteady state), typical process inputs to fermentation processes, such as cells, in the databank and roust optimization solvers that can solve the MOO objective problem usually encountered in fermentation-based biorefineries, and the ability to simulate integrated fermentation with *in situ* product recovery processes, such as adsorption and pervaporation. As a result, previous simulation efforts of integrated batch fermentations and *in situ* separations decoupled these interdependent steps with a steady state fermentation broth representing the final fermentation broth and a steady state separation of the resulting final stream. Due to the inherently non-linear nature, complexity and large number of variables involved in optimization of fermentation-based biorefineries, the available optimization tools are inadequate. For example, Aspen Plus V9 (Aspen Technology; Cambridge, Massachusetts),[29-31] can only handle SOO whereas SuperPro Designer® (Intellingen; Scott's Plain, NJ)[30, 32] has no optimization solver.

1.2 Research objectives

PSE played a critical role in the success of the refinery and chemical industries and has the potential to play a similar role in developing and expanding fermentation-based biorefineries to be economically viable. In this work, commercial PSE tools, Aspen Plus and MATLAB, are integrated on a novel platform to develop techniques and procedures to simulate time-dependent fermentation processes without and with *in situ* separations for the design, analyses, and optimization of fermentation-based biorefineries.

The batch fermentation process is simulated as an unsteady state process using the steady state process simulation platform of Aspen Plus using traditional fermentation models employing autocatalytic cell growth, and cell-dependent substrate utilization and product synthesis. The unsteady state batch fermentation is integrated with *in situ* product recovery alleviate the effect of product inhibition on the yield and productivity of the batch fermentation. The batch fermentation and integrated unsteady state batch fermentation and *in situ* separations in Aspen Plus are then linked to the robust optimization in MATLAB to optimize the operating conditions of fermentation processes. This platform of virtual experimentation through *in silico* analyses provides insights that can guide the choice of decision variables and conditions for optimum performance and as a decision-support tool for fermentation-based biorefineries.

In Chapter 2, a review of the process of converting lignocellulosic biomass to produce alternative sustainable bioproducts to fossil-based products is presented. Biorefineries are defined and classified, and the similarities and differences between a biorefinery and refinery in terms of feedstock, processing and products are elucidated. Further, the different classes of feedstock used in biorefineries are discussed together with the major advantages and disadvantages of each class of feedstock. Following the feedstock classification, the process of converting lignocellulosic biomass into biofuels via fermentation (pretreatment, hydrolysis, and fermentation) are briefly discussed. Finally, the different fermentation schemes that are used together with *in situ* product recovery to alleviate substrate and product inhibitions to microorganisms are described with examples.

In Chapter 3, the challenges that hinder the readily application of PSE tools and methods to fermentation-based biorefineries are discussed. PSE is defined and the role that PSE can play in the design, analyses and optimization of fermentation-based biorefineries are presented. Previous simulation and optimization efforts are discussed with illustrative examples to shed light on the difficulty and what is missing in using traditional process simulators for fermentation-based biorefineries. Mathematical models are central to understanding fermentation processes.[33-35] The accuracy of the results of a simulation depends in part on the quality of the mathematical model used taking into account process inputs and conditions.[36] Thus, the classification of different mathematical models based on cell population are presented with emphasizes on the commonly used models. The gap between developments in biology through genetic engineering, advanced measurement, monitoring and control tools for bioprocesses and the translation of these improvements to develop realistic mathematical models that are computationally tractable are then elucidated.

To address some of the challenges in using commercial PSE tools for fermentation-based biorefineries, a general framework that links a robust non-linear optimization solver to a traditional process simulator that has been customized to simulate unsteady state fermentation processes through a two-way communication platform is proposed. Finally, the prospects for improving the current state of bioprocess simulations and modeling are discussed. This discussion emphasizes what is required in terms of the development of mathematical models, reconciliation of *in silico* analyses results with experimentation, the adaptation that traditional process simulators need in order to expand the capabilities of unit operation models to readily simulate fermentation processes and integrated fermentation and *in situ* separations.

Chapter 4 describes the development of techniques to simulate batch fermentation processes as unsteady state using the ABE fermentation as a model system. Time-dependent fermentation models describing the batch fermentation are incorporated in a FORTRAN user kinetics subroutine and linked to the batch reactor in Aspen Plus to simulate unsteady state batch processes. This simulation approach allows the unsteady state batch fermentation to be integrated with an *in situ* product recovery via gas stripping to provide time-dependent information and separations based on thermodynamic models and phase equilibria equations. The techniques and procedures developed for the unsteady state batch fermentation simulation are foundational to the developments in Chapters 5 and 6. The simulation results of the traditional steady state approach is compared with that of the unsteady state approach developed without and with *in situ* product recovery to emphasize the importance of incorporating autocatalytic cell growth, substrate and product inhibitions, and time-dependent concentration of species on the performance of the fermentation process.

In Chapter 5, a case study that demonstrates how the novel framework proposed in Chapter 3 is used to optimize the operating conditions (gas flow rate relative to the fermentation volume and gas stripping initiation times) to maximize the total ABE produced, ABE selectivity and the concentration of ABE in the stripped stream of an integrated batch and *in situ* gas stripping process is presented. The genetic algorithm multi-objective and single-objective optimization in MATLAB is linked to the batch reactor in Aspen Plus through a Microsoft® Component Object Model (COM) interface. The batch reactor in Aspen Plus is coupled with a Fortran user kinetics subroutine (developed in Chapter 4) to evaluate the data needed to determine the objective function values in the optimization process in MATLAB. The cell growth model in the mathematical model describing the batch fermentation process is modified to use Monod kinetics to describe cell growth based on substrate utilization and incorporate product (butanol) inhibition. The new parameter introduced is evaluated by minimizing the sum of the square of the errors between four sets of experimental data and model predictions from the solution of the ordinary differential equations (ODEs) in MATLAB. The MOO and SOO optimization results are compared to reveal the trade-off and interaction that the MOO approach presents to support the understanding of fermentation processes and offer insights that may not be gained through laboratory experiments alone.

The batch fermentation is susceptible to both substrate and product inhibitions. Unlike ethanolic batch fermentation from first generations biomass, such as starch and corn, lignocellulosic hydrolysate contains a relatively lower concentration of sugars. To demonstrate the crippling effect that product inhibition has on microbial growth kinetics and the performance of the batch fermentation and offer insights about the important role the choice of design parameters and process variables play in fermentation processes, the novel platform linking the robust genetic algorithm in MATLAB to an unsteady state batch fermentation in Aspen Plus through a COM interface is presented in Chapter 6. This work builds on the foundational techniques developed in Chapters 3, 4 and 5. A time-dependent batch ethanolic model based on the utilization of both glucose and xylose with autocatalytic cell growth, substrate limitation, and substrate and product inhibitions is used. The MOO and SOO results are compared for a batch ethanolic process maximizing the total ethanol produced, ethanol productivity, ethanol yield, and the fraction of sugars converted by manipulating the initial concentration of sugars and the batch fermentation time. The ability of the unsteady state fermentation simulation in Aspen Plus to provide time-dependent information is utilized to shed light on the fermentation dynamics for cells, substrates and solvents and reveal the crippling effects of product inhibition on the performance of the batch process. Further, the enhancement in the performance of batch fermentation when it is integrated with an *in situ* product recovery is demonstrated by using an integrated batch and *in situ* gas stripping process in the MOO and SOO using techniques developed in Chapters 3, 4 and 5.

Chapter 7 summarizes the major conclusions from Chapters 3 – 6 and highlights the novelty and contributions of this work, as well as a discussion on the future directions for this work.

Chapter 2

Background

2.1 Bio-based products as sustainable alternatives to fossil-based products

The production of liquid biofuels, such as bioethanol and biobutanol, using lignocellulosic biomass serves as a sustainable alternative energy source to the depleting fossil fuels[1] (four barrels of oil are consumed for every barrel of oil discovered[37]), helps reduce environmental pollution and is not subject to the effects of instable political situations in some oil producing countries.[38-39] The United States has the capacity to produce 1.6 billion dry tons of biomass annually to produce bioenergy and bioproducts in a sustainable way while still meeting food, feed and fiber demand. Additionally, the U.S. potentially could produce 85 billion gallons of biofuels every year to replace about 30% of the nation's current fossil fuels[39] by 2030.[40] Biobutanol and bioethanol have properties that make them excellent biomass-derived liquid transportation fuels. For instance, the net heat of combustion of butanol and ethanol are 83% and 65%, respectively, of that of gasoline (32.5 MJ/L).

The solubility of butanol in water is 7.7 g/100 mL (at 20^{o}C) which make butanol very hydrophobic whereas ethanol is miscible in water. Butanol has a lower vapor pressure and volatility, higher boiling point (117.7^{o}C) and flash point (29^{o}C); butanol is relatively safer to handle than other alcohols.[37-38] The Reid's value, a measure of the tendency for a fluid to evaporate, for butanol, gasoline and ethanol are 0.33, 4.5 and 2.0 psi, respectively. Butanol can replace and/or be blended with gasoline for use in existing car engines; 85% butanol/gasoline blend can be used in existing engines.[37] Butanol and its gasoline blend at any concentration can therefore be stored and transported in existing storage and distributing facilities.[4, 38, 41] The United States Environmental Protection Agency (EPA) approved the use of 15% ethanol/ gasoline blend for all vehicles made from the year 2001 onward.[39] The solvents produced from lignocellulosic processing are used rubber monomers, butadiene and dimethyl butadiene production.[42] Consequently, a lot of research interest and efforts are invested in developing new processes and improving existing processes to convert biomass into sustainable biofuels.[43]

2.2 Biorefinery

Global politics, environmental concerns, and geographical drivers are constantly driving the development of bioproducts from biorefineries. Concerns over the use of food and feed as feedstock in biorefineries are increasingly rising, driving lignocellulosic biomass as the expected most important source of biomass in the future. Transportation fuels are the immediate focus of biorefineries, where these biofuels are blended with their fossil fuel based counterparts, such as gasoline, diesel, and natural gas. Similar efforts (many decades) invested in the research, design and optimization of refineries are expected to be applied to expand and sustain the biorefinery industry, drawing on the expertise from chemistry, catalysis and the engineering community. It is expected that biorefineries are integrated with existing refinery processes and infrastructure to be economically viable. For example, the existing refinery infrastructure in the transportation industry could be used for the transportation fuel from biorefineries.[44]

2.2.1 Biorefinery definition

Several definitions of a biorefinery are used depending on the context and application. One definition of a biorefinery is "a facility that integrates conversion processes and equipment to produce fuels, power and chemicals from biomass."[45] A biorefinery is also defined as "the sustainable processing of biomass into a spectrum of marketable products and energy."[44] By the latter definition, a biorefinery could be a facility, process, plant or collection of facilities. The marketable products include food, feed, materials and chemicals whereas fuels, power, and heat constitute the energy products.[44]

2.2.2 Biorefinery classification

A biorefinery can be classified based on the maturity of the processing technology, type of feedstock used, conversion process, and the intermediate products as shown in Figure 2.1, which complements Figure 3.1 in Chapter 3.

2.2.3 Similarities and differences between a biorefinery and refinery

The concept of a biorefinery is similar to that of a refinery in that they both use processing technologies and equipment to produce fuels, energy and chemicals as illustrated in Figure 2.2. A biorefinery is often compared to a refinery in terms of the feedstock, building block platforms, processing, and the chemical intermediated that are at the commercial scale to show the similarities and differences between a refinery and biorefinery as shown in Table 2.1

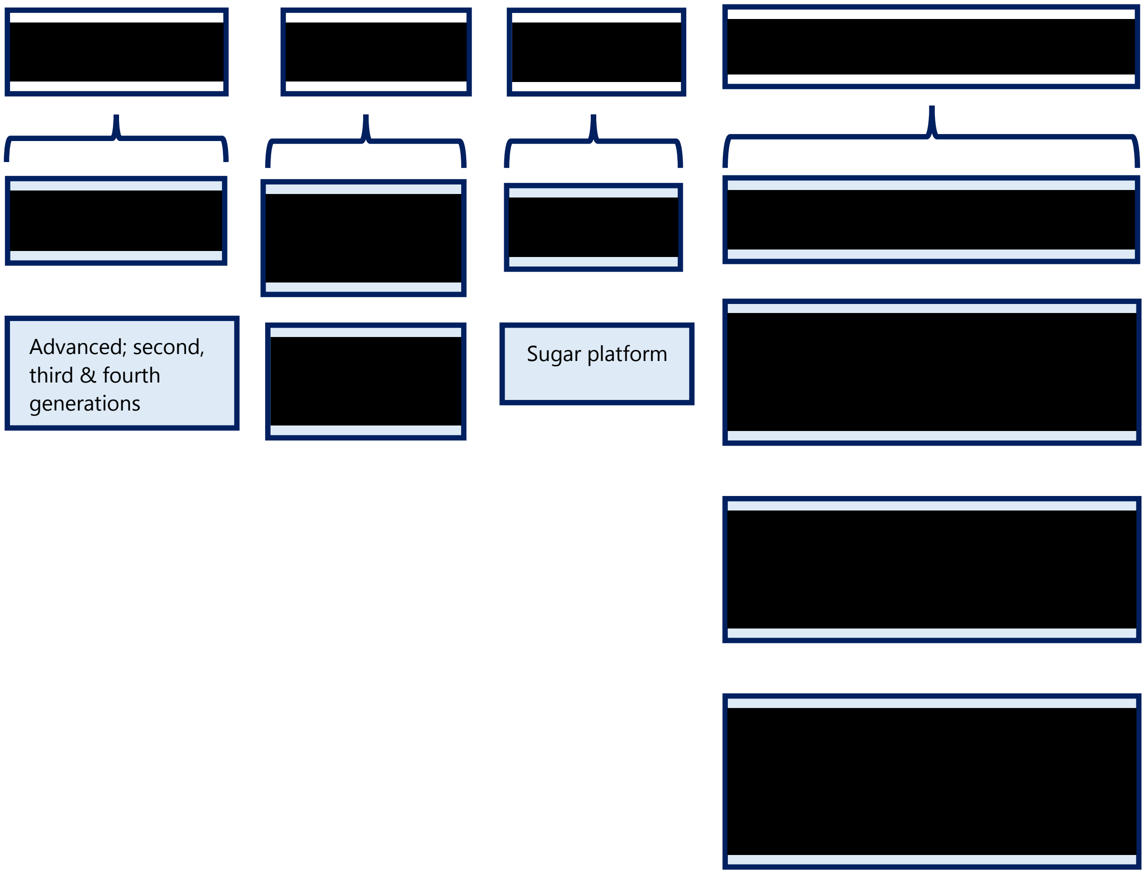


Figure 2.1 Flowchart showing the classification of biorefineries based on the maturity of the technology process, conversion process, type of intermediate product, and type of feed stock used.[44]

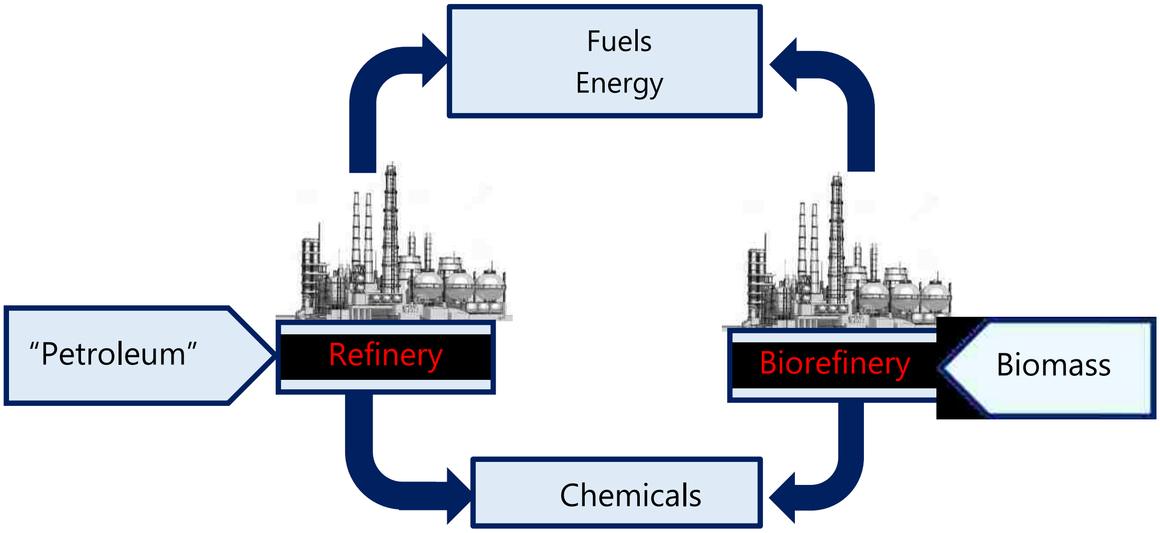


Figure 2.2 The similarities between the concept of refineries and biorefineries.[45]

Table 2.1 Comparison of refineries and biorefineries based on the feedstock, building block chemical composition, processing and building block intermediates produced on the commercial scale.[44]

	Refinery	Biorefinery
Feedstock	Relatively homogeneous	Heterogeneous e.g. carbohydrates, lignin, proteins, oils, ash Polymeric feedstock e.g. cellulose, starch, proteins.
	Low oxygen content. E.g. Crude oil: C, H, O has composition of 85 – 90, 10 – 14, and 0 – 1.5 %, respectively.	High oxygen content. E.g. Wood: C, H, O has composition of 50, 6, and 43 %, respectively.
	Weight of product (mole/mole) increases with processing	Weight of product (mole/mole) decreases with processing
	Significant sulfur content (very high sometimes)	Low sulfur content with high inorganic content. E.g. silica
Building blocks	Ethylene, propylene, methane, benzene, toluene, xylene isomers	Glucose, xylose, fatty acids (e.g. oleic, stearic, sebacic).
(Bio)chemical	Essentially a chemical process	Chemical, biotechnology, and biochemical
	Heteroatoms introduction E.g. O, N	Removal of oxygen
	Relative homogeneous process to form building blocks. E.g. steam cracking, catalytic reforming	Relative heterogeneous process to form building blocks
	Multiple conversion chemistries used	Narrow range of conversion chemistries such as dehydration, fermentation, hydrogenation
Intermediate products at the commercial scale	Several	Relatively fewer compared to refineries. E.g. ethanol, biodiesel, furfural, etc.

2.3 Classification of the feed stock for biorefineries

2.3.1 First generation biomass

First generation biomass consists of edible food crops such as corn, sugarcane, whey, barley, potatoes, sugarcane beets, etc.[46-47] The use of edible biomass as feedstock in most commercial biorefineries have been met with a lot of resistance and concerns as these are used as food and feed.

2.3.2 Second generation biomass

Lignocellulosic biomass as a second generation biomass uses low cost, abundant and non-edible feedstock and reduces the feud between fuel and food competition associated with first generation biomass that are edible feedstock. [37] Lignocellulosic biomass comprises of non-edible residues from food crop production (corn stover, wheat and rice straw[41]), forest residues and energy crops (switchgrass, miscanthus)[39, 41] as well as waste such as municipal solid waste, etc. The major challenges in using lignocellulosic biomass in a traditional biorefinery to produce biofuels are the seasonal, annual variability in biomass supply[39] and the need to harvest, transport and store voluminous quantities of biomass feedstock to produce a continuous supply of fuel.

2.3.3 Third generation biomass

Algae (microalgae and macroalgae, Figure 2.1) constitute the feedstock of third generation biomass. The disadvantage of third generation biomass relate to the high amount of water required in the processing and the high water content of lipids extracted from algae, requiring dewatering.[46, 48]

2.3.4 Fourth generation biomass

Fourth generation biomass uses metabolic engineering to produce engineered algae with superior characteristics for biofuel production. Biofuel production form fourth generation biomass may be economically viable in the long-term compared to third generation biomass. The development of cost effective photo reactors and efficient separation technologies would decrease the capital cost significantly.[48]

2.4 The process of converting lignocellulosic biomass into bioproducts via fermentation

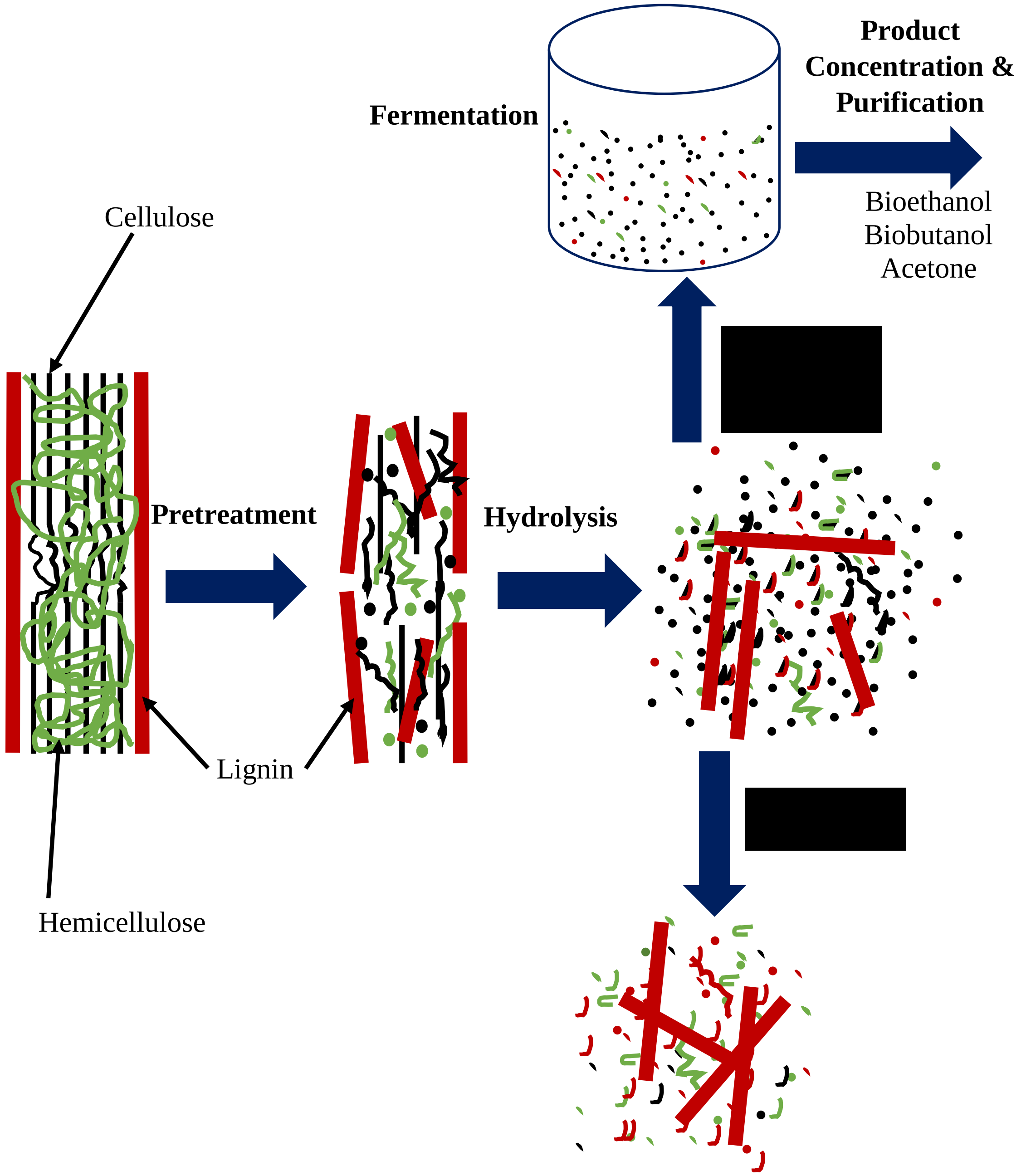
Lignocellulose biomass is made up of cellulose microfibrils (35 - 50% of the total dry mass)[49-51], lignin (15 - 25%), and hemicellulose (23 - 32%).[51] The biochemical conversion process of lignocellulosic biomass into bioproducts involves pretreatment, hydrolysis, fermentation, product concentration and recovery[39, 52] as shown in Figure 2.3

2.4.1 Pretreatment

The goal of pretreatment is to overcome the recalcitrance of lignocellulosic biomass and make the polymeric carbohydrates (cellulose polymers) more accessible to enzymes or chemicals used in the hydrolysis process to release fermentable sugars for fermentation[53] as shown in Figure 2.3. Pretreatment methods include: physical (milling and grinding), physicochemical (steam pretreatment/autohydrolysis, hydrothermolysis, and wet oxidation), chemical (alkali, dilute acid, oxidizing agents, and organic solvents), biological (fungal), electrical, or a combination of these.[53]

2.4.2 Hydrolysis

Following pretreatment, the polymeric carbohydrates (cellulose and hemicellulose polymers) are broken down (hydrolyzed) into fermentable sugar monomers (such as glucose and xylose) using biological or chemical hydrolysis.[54] The microorganisms (bacteria and fungi) used in biological hydrolysis are able to produce enzymes for cellulose hydrolysis. Examples of enzymes for hydrolysis include cellulases and hemicellulases.[55] Complexed cellulases are often produced by anaerobic bacteria such as *Clostridium thermocellum.*[56] *C. thermocellum* is a thermophilic anaerobic bacterium that catabolizes cellulose and other carbohydrates by producing cellulosomes (extracellular multi-enzymes).[57] The cellulosomes may be attached to the cell (cell-associated) or free forms. Chemical hydrolysis process include acid, alkaline and oxidative delignification. Concentrated acid (sulfuric and hydrochloric acids) hydrolysis are very effective but concentrated acids are toxic, hazardous and corrosive. Dilute acids of sulfuric and hydrochloric acids hydrolysis are often used at moderate and high temperature. Alkaline hydrolysis frequently uses dilute ammonia whereas oxidative delignification uses hydrogen peroxide.[58]



[redacted]

[redacted]

Figure 2.3 Schematic of the pretreatment and hydrolysis of lignocellulosic biomass to produce fermentable sugars.[53]

2.4.3 Fermentation

"Fermentation is a biological process where a substrate is converted into a valuable product by a microbial organism."[2] Hydrolysis produces both hexoses (glucose, galactose, and mannose) and pentoses (xylose and arabinose). One of the challenges in fermentation is the ability of microorganisms to, at least, use the major sugars (glucose and xylose) from lignocellulosic biomass. Various microorganisms have been modified through genetic engineering to utilize both hexoses and pentoses. Examples include yeast (e.g., *Saccharomyces cerevisiae*, *Scheffersomyces (Pichia) stipites, Kluyveromyces marxianus)* and bacteria (e.g., *Zymomonas mobilis, Escherichia coli, Klebsiella oxytoca*). The challenge still remains in developing these engineered strains further to overcome the incomplete utilization of pentose sugars, low productivities, yields and titers, and inhibition to products generated in the processing (e.g. acetic acid and furfural).[59]

The Acetone-butanol-ethanol fermentation. The biochemical fermentation process that produces biobutanol using microorganisms is known as acetone-butanol-ethanol (ABE) fermentation because a mixture of acetone, butanol and ethanol are produced. A product mix of 6:3:1 of butanol, acetone, and ethanol, respectively is typical produced.[38] The most common microorganisms used for ABE fermentation, anaerobic bacteria such as solventogenic *Clostridia,* [38]are a heterogeneous collection of gram-positive, non-sulfate-reducing, spore-forming, rod-shaped bacteria.[60] Examples of solventogenic *Clostridia* are: *Clostridium acetobutylicum, C. beijerinckii, C. saccharobutylicum. Saccharoperbutylacetonicum.*

ABE producing *Clostridia* produces acetone-butanol-ethanol (ABE) during the conventional batch fermentation using a carbon source through two separate growth phases: the exponential acidogenic phase (butyric acid, acetic acid, carbon dioxide and hydrogen are produced) and late exponential and stationary solventogenic phase[4, 38] (the excreted acids are taken up and converted to acetone and butanol in a typical ratio of 2:1)[4] as shown in Figure 2.4. The switch from the acidogenic to the solventogenic phase is to avoid death because of the lowering of the pH of the fermentation broth by the acids produced.

2.5 Fermentation design strategies

Different fermentation schemes are used to circumvent product and substrate inhibitions. These include batch, fed-batch, continuous and process integration as a combination of batch, fed-batch or continuous fermentation and product recovery techniques.[1]. In a batch fermentation, the substrate, nutrients and microorganisms are charged to a bioreactor for fermentation[42] as shown in Figure 2.5A. To reduce substrate inhibition, fed-batch fermentation is used; a bioreactor is charged with a relatively low concentration of substrate and as the substrate is used up, fresh substrate is supplied at a rate that keeps the substrate concentration in the reactor below the inhibitory levels to the microorganisms,[10] Figure 2.5B.

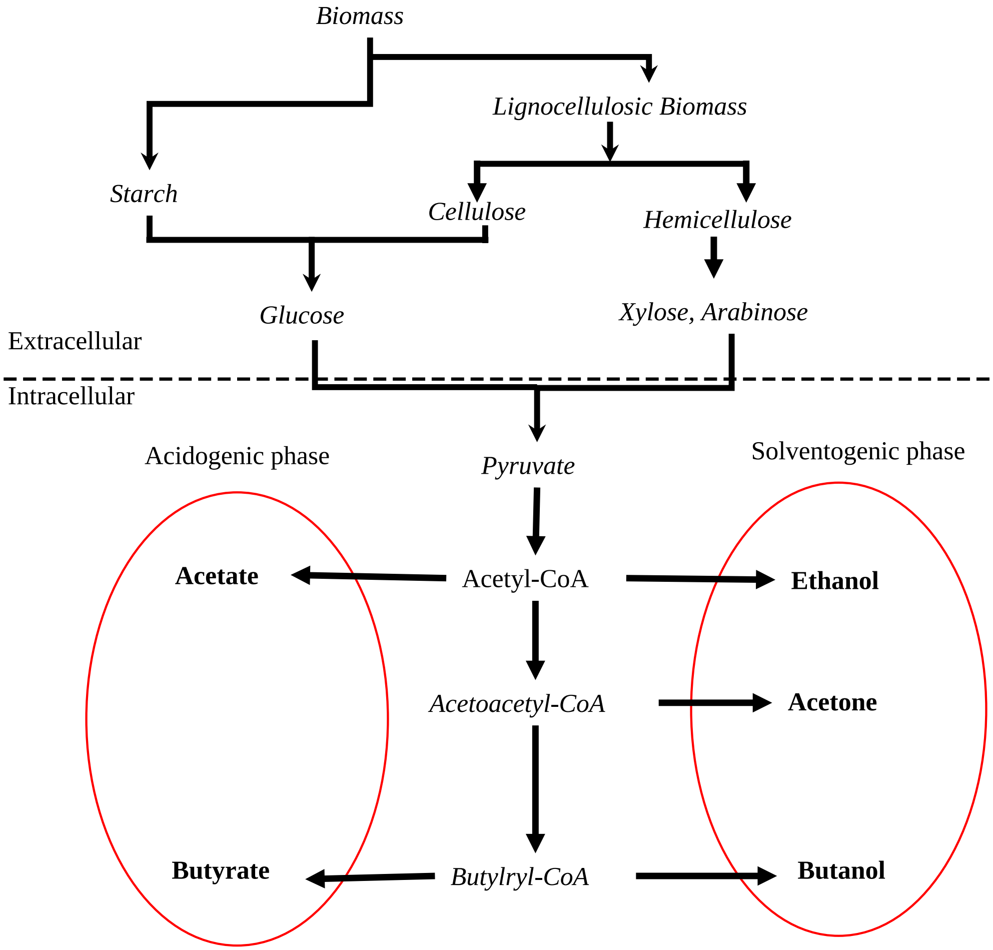


Figure 2.4 Schematic of the metabolism of biomass by solventogenic Clostridia.[5, 61]

To circumvent product inhibition, the batch and fed-batch fermentations are integrated with *in situ* product recovery techniques such as gas stripping, adsorption, pervaporation, etc. as shown in Figure 2.5C. The accumulation of toxic products have adverse effect on microorganisms in fermentation. For example, Ezeji, *et al.*[1] showed that fermentation in a combined fed-batch and *in situ* gas stripping process ceased after 201 h due to the accumulation of salts, dead cells and unknown bioproducts.[1] The ideal process design configuration is an integrated continuous production process combining continuous fermentation, product recovery and "bleeding" (taking out portions of the fermentation broth intermittently to eliminate the accumulation of toxic products[1] as shown in Figure 2.5D.

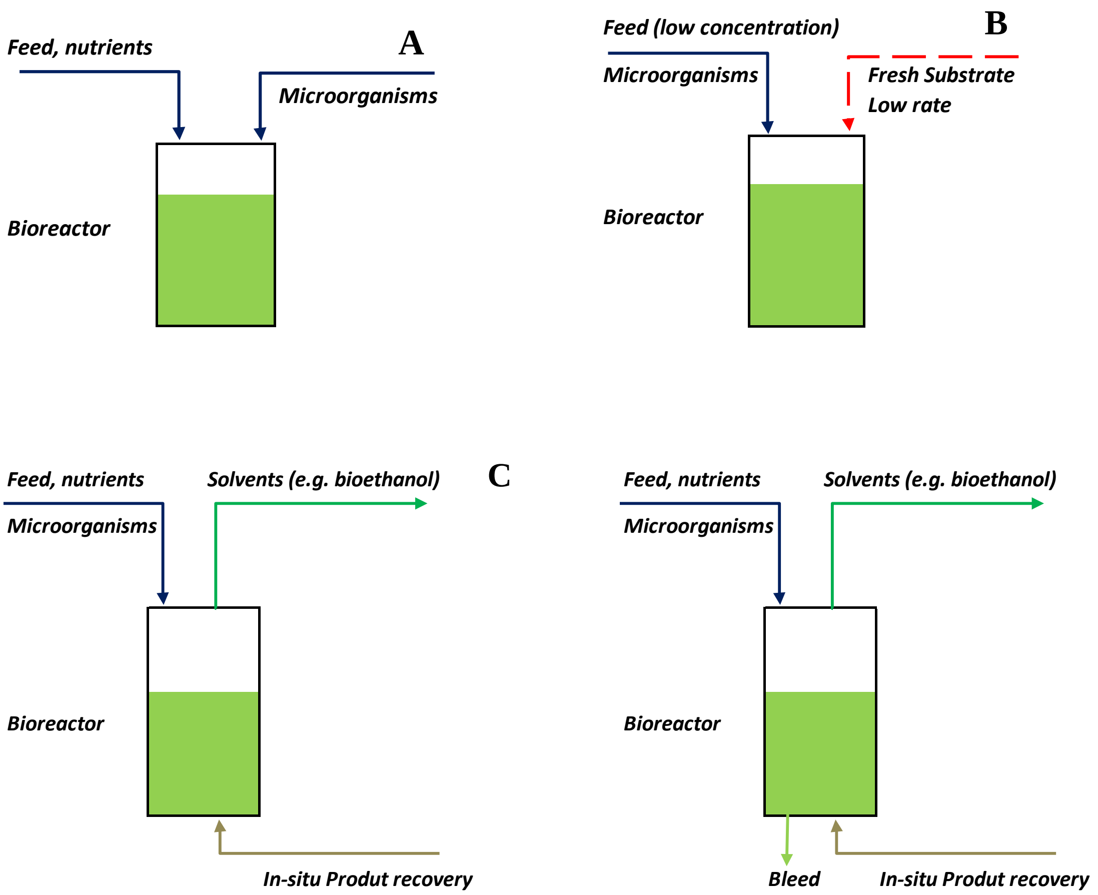


Figure 2.5 Schematic of different fermentation schemes
A) batch B) fed-batch C) integrated fermentation with in situ product recovery D) integrated fermentation and in situ product recovery with bleeding.

2.5.1 Gas stripping

Gas stripping selectively removes volatile solvents from the aqueous fermentation broth by continuously feeding a stream of gas. Gas stripping is more economical because it has the option to use fermentation product gases (hydrogen and carbon dioxide)[10] or an inert gas as the stripping gas, the existing fermentation set-up is used with minimal modifications and no additional chemicals are used.[38] The existing gases from the fermentation broth contain solvents (e.g. acetone, butanol, and ethanol) at their equilibrium partial pressures and are typically partially condensed in a condenser to recover the solvents in the gas stream. The stripped gas is then recycled continuously through the fermentation broth.[38]

2.5.2 Adsorption

Adsorption selectively adsorbs butanol or ABE products onto an adsorbent and then temperature change[38] (thermal-swing-adsorption [62]) or a displacer is used to desorb the products. This process establishes a dynamic equilibrium for the distribution of the solute, species being adsorbed (butanol, ABE), between the fluid and the adsorbent surface.[62] The equilibrium relation is usually expressed as a function of the concentration of the adsorbed solute (adsorbate) in the liquid mixture and the amount of solute adsorbed on the solid material (adsorbent), expressed as the mass, moles, or volume of the adsorbent per unit mass or surface area of the adsorbent. A plot of the solute loading on the adsorbent versus the equilibrium concentration of the solute in a fluid is called an adsorption isotherm. This equilibrium isotherm governs the extent to which a solute can be adsorbed from a given liquid mixture on an adsorbent for a given set of conditions. The maximum capacity of an adsorbent can thus be estimated, for example from a Langmuir isotherm model, for a given adsorbent for the adsorption of a solute from a liquid mixture.[62] The selection of the effective adsorbent is based on the nature of the adsorption isotherm. Isotherms with a downward curvature are called favorable isotherms. A highly favorable adsorbent is effective in separating dilute solutions but will be strongly unfavorable in the desorption of the components adsorbed.[63]

The type of adsorbent used depends of the adsorption rate, adsorption capacity, desorption rate, affinity for the product of interest and the cost of the adsorbent. Adsorbents with high adsorption rate give fast kinetics that allow for high circulation flow rate. A larger quantity of adsorbent is required when an adsorbent with slow adsorption rate is selected. A good adsorbent should be effective in selectively adsorbing the desired product and regenerated for reuse a large number of times.[38] The most common adsorbents used for alcohol separation are activated carbons (ACs), polymeric resins, polyvinyl pyridine (PVP) and zeolites.[38] Generally, adsorbents are highly porous with surface areas typically of several hundred square meters per gram.[63] ACs are used to adsorb nonpolar molecules[62-63] and weakly polar molecules, especially hydrocarbons, because ACs have nonpolar surfaces, and a high specific surface area and are hydrophobic.[62] Zeolites have crystalline aluminosilicates with specific pore sizes located within small crystals and are thus able to separate by molecular shape, size and polarity.[64]

2.5.3 Pervaporation

Pervaporation as a membrane separation technique involving the partial vaporization of a liquid mixture through a membrane that is selective to one or more of the components of the mixture. Pervaporation uses vacuum or a sweep gas at the permeate side to recover the permeate from a liquid feed mixture that is in contact with the other side of the membrane. A cold trap is used to recover the permeate vapor. Pervaporation has the advantages of having a low energy requirement, prevents the loss of nutrients and substrate and potentially has no effect on the microorganisms used in fermentation. However, pervaporation suffers from low permeation flux, fouling and usually requires large membrane surface area for effective separation.[38]

2.6 Application of Process Systems Engineering to fermentation-based biorefineries

Process Systems Engineering uses computer-based tools and methods to design, analyze, optimize, and control complex processes that may involve a physical, chemical and/or biological change. PSE can be described as a virtual laboratory where “experiments” are conducted *in silico*. Through such a virtual platform, a large number of process variables, such as process inputs and conditions, and an overall process-wide view can be simulated to give a global perspective of the process at hand. A real process is translated from a laboratory experiment to the virtual realm through PSE as illustrated with a fermentation process in Figure 2.6. For a fermentation process, the basic engineering principles and the “theory”, such as Monod kinetics to describe cell growth, are formulated and used to translate the fermentation process into a mathematical model (equations). In the formulation of the mathematical model, mathematical equations, for example, ODEs, are used to describe the rate of substrate utilization, autocatalytic cell growth and product formation. Typically, the model parameters are fitted by statistical methods. For example, an optimization process can be used to fit the model parameters, where the sum of the squared differences between the model predictions and experimental data is minimized.

Using the mathematical model developed, the simulation predictions of the process from a process simulator can be compared to the experimental data. Depending on the range of applicability of the conditions and inputs used to formulate the model, the model predictions can be used for interpolation and/or extrapolation to gain more insight about the process at hand. In this way, the use of PSE tools and methods help to reduce unnecessary experimentation, cut down cost and save resources and time to know the most informative experiments for process design, control and optimization.

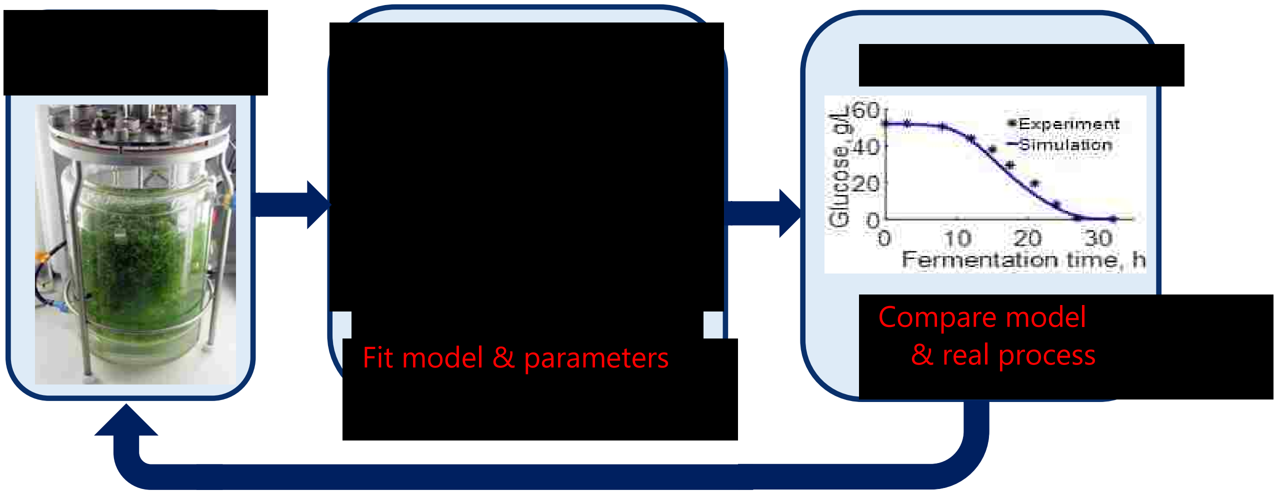


Figure 2.6 The process of translating a real process into a mathematical model using Process Systems Engineering tools and methods.

Unlike the traditional chemical and refinery processes, PSE tools and methods are not routinely applied for the design, analysis and optimization of fermentation-based biorefineries. When a refinery and a biorefinery are compared based on the nature of the processes, components involved and the mode of operation, the reasons why it has been difficult to apply traditional PSE tools and methods to biorefineries relative to a refinery can be summarized in Figure 2.7. A refinery process mostly involve a physical and/or chemical process and is relatively easier to be described with mathematical models compared to a biorefinery process, which is a biological process. Fermentation-based biorefineries are particularly challenging to model with traditional PSE tools when fermentation is integrated with *in situ* product recovery and separation techniques such as adsorption and membrane separation.

As shown in Figure 2.7, a refinery typically involves hydrocarbon components (hydrocarbons are easier to characterize with the corresponding physical property data readily available in commercial process simulators) and these components are usually found in the database of commercial process simulators. On the contrary, a fermentation process involve components such as microorganisms (e.g. yeast), molasses and bagasse. These components are either hard to characterize or have no defined composition, resulting in these components not found in the databank of process simulators because of missing and/or incomplete physical property data. Apart from the start-up and shutdown processing steps, refinery processes are mostly operated as steady state whereas fermentation-based biorefineries are frequently operated as batch and semi-batch processes. Additionally, fermentation processes are often described and modeled with batch data even when fermentation processes are operated as continuous processes. Batch and semi-batch processes are inherently unsteady state; the properties of the system, such as composition and concentration, change with time. This time-dependent nature of fermentation processes render traditional steady state process simulators inadequate to describe and simulate the inherently unsteady state fermentation process.

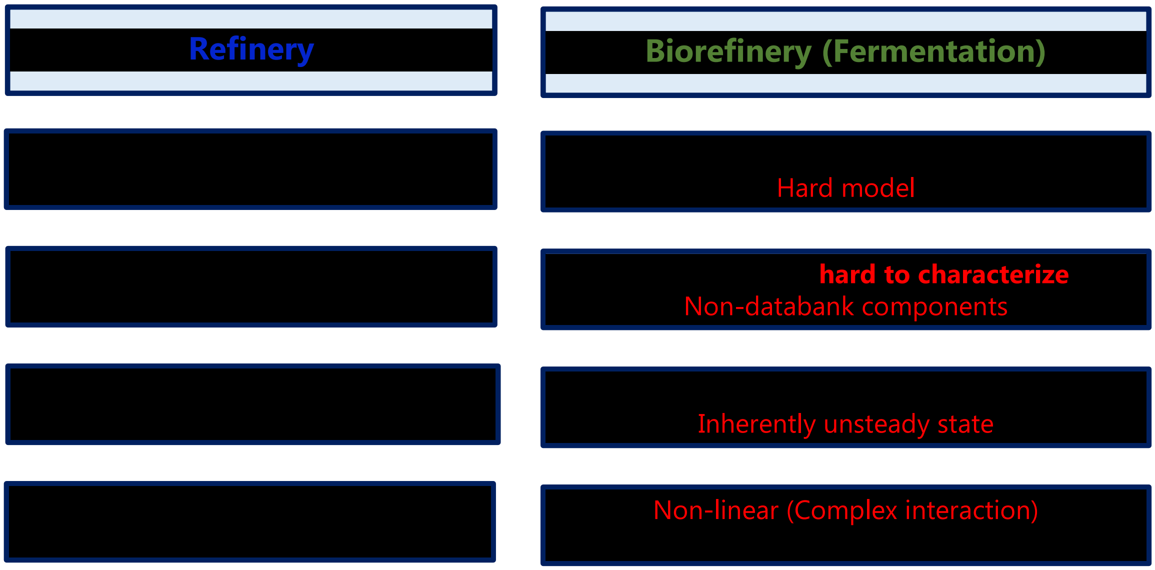

Refinery
Biorefinery (Fermentation)
Hard model
hard to characterize
Non-databank components
Inherently unsteady state
Non-linear (Complex interaction)

Figure 2.7 Comparison of a refinery with a biorefinery to highlight the challenges for applying commercial process simulators to biorefinery processes.

Optimization of refinery processes typically involve linear mathematical models for which shortcut/single optimization approaches available in most commercial process simulators are adequate. In contrast, fermentation-based biorefineries involve the complex interaction of microorganisms and unsteady state fermentation environment, resulting in non-linear mathematical models. Fermentation processes, therefore, require rigorous optimization approaches, such as multi-objective optimization, which are often not available in commercial process simulators.

In Table 2.2, the traditional steady state process simulator, Aspen Plus, is used as a model process simulator to further elucidate the challenges hindering the readily application of commercial PSE tools to fermentation-based biorefineries. In Aspen Plus V9, typical fermentation components such as biomass and microorganisms are non-databank components. In addition, the typical ODEs describing fermentation processes do not conform to the form of the in-built power law rate-based kinetic model in Aspen Plus V9. Aspen Plus V9, for example, can only handle single-objective optimization and not the typical multi-objective optimization problem that a fermentation-based biorefinery presents.

Table 2.2 Using Aspen Plus as a model process simulator to highlight the challenges for applying commercial process simulators to fermentation processes.

Process simulator	*Fermentation components in database*	*Built-in rate based kinetics*	*Optimization tools available*
Aspen Plus V9	Nondatabank: microorganism (e.g. yeast), biomass	Power Law Model: $r = kT^n exp(-E/RT) \prod (C_i)^{\vartheta_i}$	Only single objective optimization
Fermentation simulation challenges	Nondatabank: Microorganisms such as yeast and cells	Typical ODEs describing batch fermentation does not conform to the built-in rate kinetics. E.g. $\frac{dC_{BA}}{dt} = k_5 C_S \frac{K_I}{K_I + C_B} C_X - k_6 \frac{C_{BA}}{K_{BA} + C_{BA}} C_X$	Both single and multi-objective optimizations

where *r* is rate of reaction, *k* is pre-exponential factor, *T* is temperature, *n* is temperature exponent, *E* is activation energy, *R* is universal gas law constant, ϑ is concentration exponent, *i*, *is* component index, π is the product operator, *C* is component concentration, *BA* is butyric acid, *S* is glucose, *X* is cells, *B* is butanol, k_5, K_I *and* K_{BA} are parameters.

To circumvent these challenges, a commercial process simulator must be linked with an external robust optimization platform to facilitate the exchange of data between the external optimization software and the process simulator as shown in Figure 2.8. Furthermore, nondatabank components in the commercial process simulator may be incorporated as user-defined components. The non-standard ODEs describing the fermentation process can be incorporated by linking the process simulator to either an external program that can solve ODEs or user subroutine to simulate the fermentation process as an unsteady state process. Many of the optimization problems in fermentation-based biorefineries result in a multiple objective system in which some or all of the objectives are competing. As shown in Figure 2.9 for a multiple objective optimization problem, the traditional single-objective optimization (SOO) approach either designates one objective as the main objective function while other objective functions are transformed into additional constraints or a relative average sum of all the objective functions into one objective using relative weights. The SOO finds one optimal decision variable with one set of the corresponding objective function values. On the contrary, a multi-objective optimization simultaneously optimize multiple objectives to find a set of equally optimal decision variables and their corresponding sets of objective function values, collectively called Pareto optimal solutions. The Pareto optimal solutions are typically scored and ranked using the preferences and inputs of the decision maker based on algorithms such as the Net Flow Method and Rough Set Method.

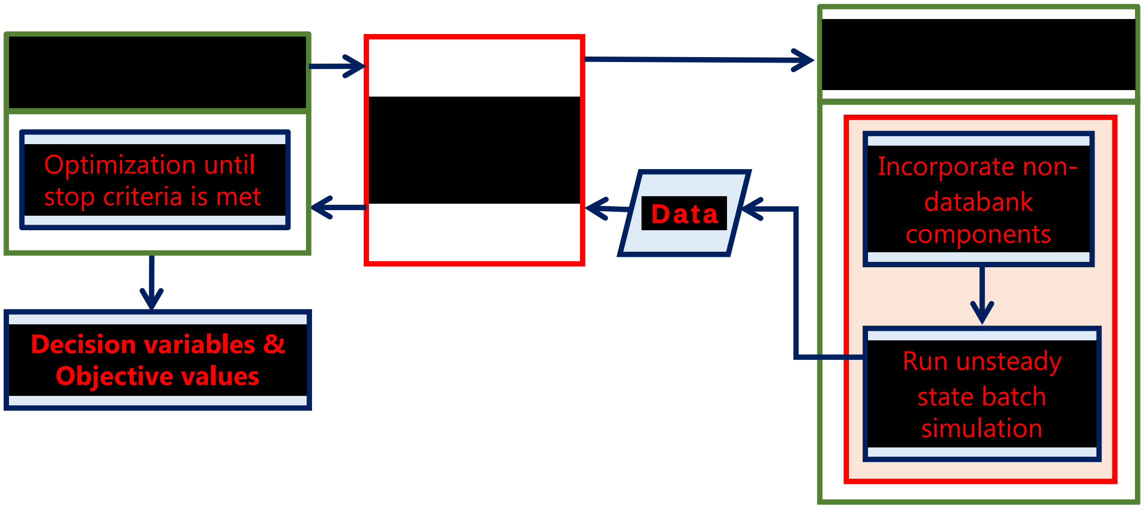

Optimization until stop criteria is met
Data
Incorporate non-databank components
Run unsteady state batch simulation
Decision variables & Objective values

Figure 2.8 General framework for optimization of fermentation process based on unsteady state simulations in commercial process simulators coupled with rigorous optimization.

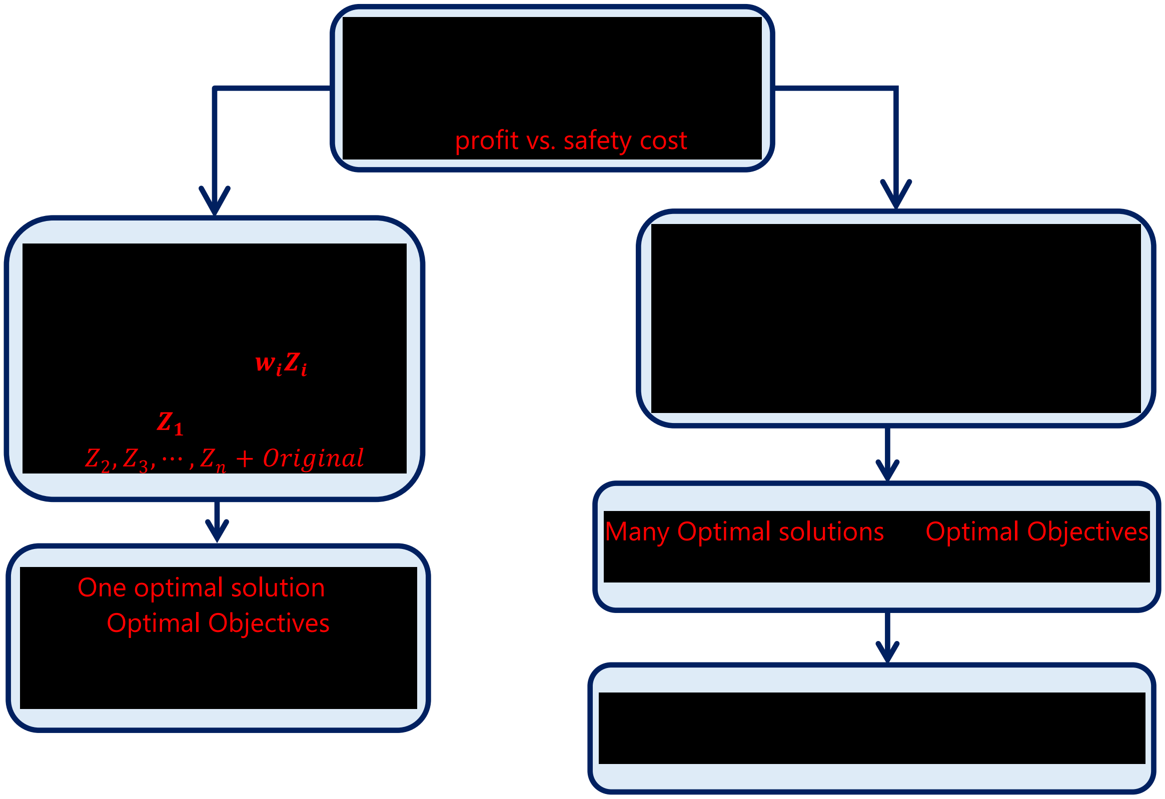


Figure 2.9 Single-objective versus multi-objective optimization approaches.

Chapter 3

A Perspective on Challenges and Prospects for Applying Process Systems Engineering Tools to Fermentation-Based Biorefineries

3.1 Summary

Unlike traditional chemical and petrochemical processes, Process Systems Engineering (PSE) tools and methods have not been routinely applied for the design, analysis, and optimization of fermentation-based biorefineries. This has greatly limited the ability to analyze these biomass-based processes, thus also limiting the potential profitability of biorefinery processes. This Perspective elucidates the challenges currently hindering the application of traditional PSE tools and methods to fermentation-based biorefineries. The current state of process simulators and mathematical models, the backbone of PSE, as applied to fermentation-based processes are addressed. Further, this contribution includes a proposed framework that can be applied to fermentation-based processes using existing process simulators with an illustrative case study to highlight how imperative PSE tools and methods are to the advancement of biorefineries. Finally, the future needs and prospects for using traditional PSE tools and methods for the design, analysis, and optimization of fermentation-based biorefineries to provide sustainable alternatives to existing processes and fossil-based products are discussed.

3.2 Introduction

Unlike traditional petrochemical products and processes, fuels and chemicals produced via fermentation have not yet benefited from decades of refinement and optimization using the available Process Systems Engineering (PSE) tools. As a result, the economic viability of these processes is often understated. Further, when relying on laboratory experimentation and pilot studies alone, finding optimal process configurations can be prohibitively time consuming and expensive. That said, biomass – particularly waste biomass from crop residues – presents a potentially sustainable source of fuels and chemicals. According to the 2017 BP Statistical Review of World Energy, production of biofuels grew at an average rate of 15.2% per annum in the United States and 14.1% per annum globally.[65] This growth rate is significant and highlights the importance of applying the tools and methods of PSE to optimize the production of biofuels. Unfortunately, creating process simulations of fermentation-based products is especially challenging because the fermentation process involves microorganisms that are difficult to characterize and model mathematically.[35] On the other hand, traditional PSE tools partially or fully lack features needed to simulate fermentation processes. Without robust computer models, optimization is difficult if not impossible.

In an era of increasing concern regarding the effects of petroleum-based products on the global climate, fuels and chemicals produced from nonfood, lignocellulosic biomass sources may provide a sustainable alternative. These so-called second generation biofuels have the potential to reduce greenhouse gas emissions. When paired with governmental incentive programs, the production of biomass-based products globally has increased dramatically.[65] However, there are numerous technical challenges that must still be addressed before processes based on lignocellulosic biomass are economically viable. These include supply chain constraints, overcoming the recalcitrance of lignocellulosic biomass to produce fermentable sugars, the effects of sugar composition and concentrations on microorganisms, and the fermentation process; as well as substrate and product inhibitions. Additionally, second generation biofuels are subject to additional uncertainties due to the upstream costs of agricultural production.[66] The experimental burdens to address these challenges are high. Therefore, if robust mathematical models can be developed, the tools and methods of PSE can be brought to bear to provide direction for experimental work. Mathematical models allow researchers to quickly assess a proposed fermentation-based biorefinery in terms of sustainability metrics (potential profitability and environmental and societal impacts).[67-68]

PSE relies heavily on computer tools and methods to analyze groups of individual unit operations as a process. Taking this broader view allows design engineers to achieve insight into the behavior of the process that is not possible when taking a unit by unit view. However, taking this approach requires the development of sophisticated mathematical models for each part of the system. This is particularly difficult for fermentation-based processes. Not only are most fermentation processes unsteady state, but the kinetics of converting raw materials, such as biomass into fuel and chemical products, must be coupled with the lag, autocatalytic growth, and death stages of the microorganism being used to carry out the chemical conversion. Currently, there is a lack of existing unit operation models in commercially available simulation packages for fermentation processes that fully include both reaction kinetics and microorganism life cycle considerations. This is in sharp contrast to petrochemical processes, which can easily be described mathematically with physical and thermodynamic models.

To this end, this work aims to provide analysis of some of the key issues hindering the use of PSE tools and methods through commercial process simulators for fermentation-based biorefineries. The challenges in applying traditional PSE tools are addressed based on mathematical models, process simulation and optimization with illustrative examples of the current state-of-the-art simulation and optimization approaches. A proposed framework that utilizes traditional process simulators is used in a case study (presented separately in Chapter 5) to highlight the important role that PSE tools play in guiding and supporting the understanding of experimental work to provide process insights that will be too difficult, expensive, or impossible through experiments alone. The prospects for application of PSE tools to fermentation-based biorefineries are also addressed, including improvements and developments in mathematical models, process simulation, and optimization needs.

3.3 Background

3.3.1 Biorefining

A biorefinery uses biomass to produce fuels, power, and chemicals in a facility that combines biomass conversion technologies and equipment.[22-26] A biorefinery can be broadly classified mainly into biochemical and thermochemical conversion platforms.[37, 39, 52, 69] The thermochemical conversion process uses heat and catalysts (using gasification or pyrolysis).[39] As shown in Figure 3.1, the biochemical platform, also known as the sugar platform, depending on the biocatalysts (microorganisms and enzymes)[52] used in fermentation, can convert biomass into biofuels (e.g., bioethanol and biobutanol) and chemicals (e.g., lactic and succinic acids). “Fermentation is a biological process where a substrate is converted into a valuable product by a microbial organism.”[2] Biomass is converted to biofuels in the sugar platform in four steps: pretreatment, hydrolysis of pretreated material to monosugars (fermentable sugars),[39] fermentation of monosugars and purification.[70-71]

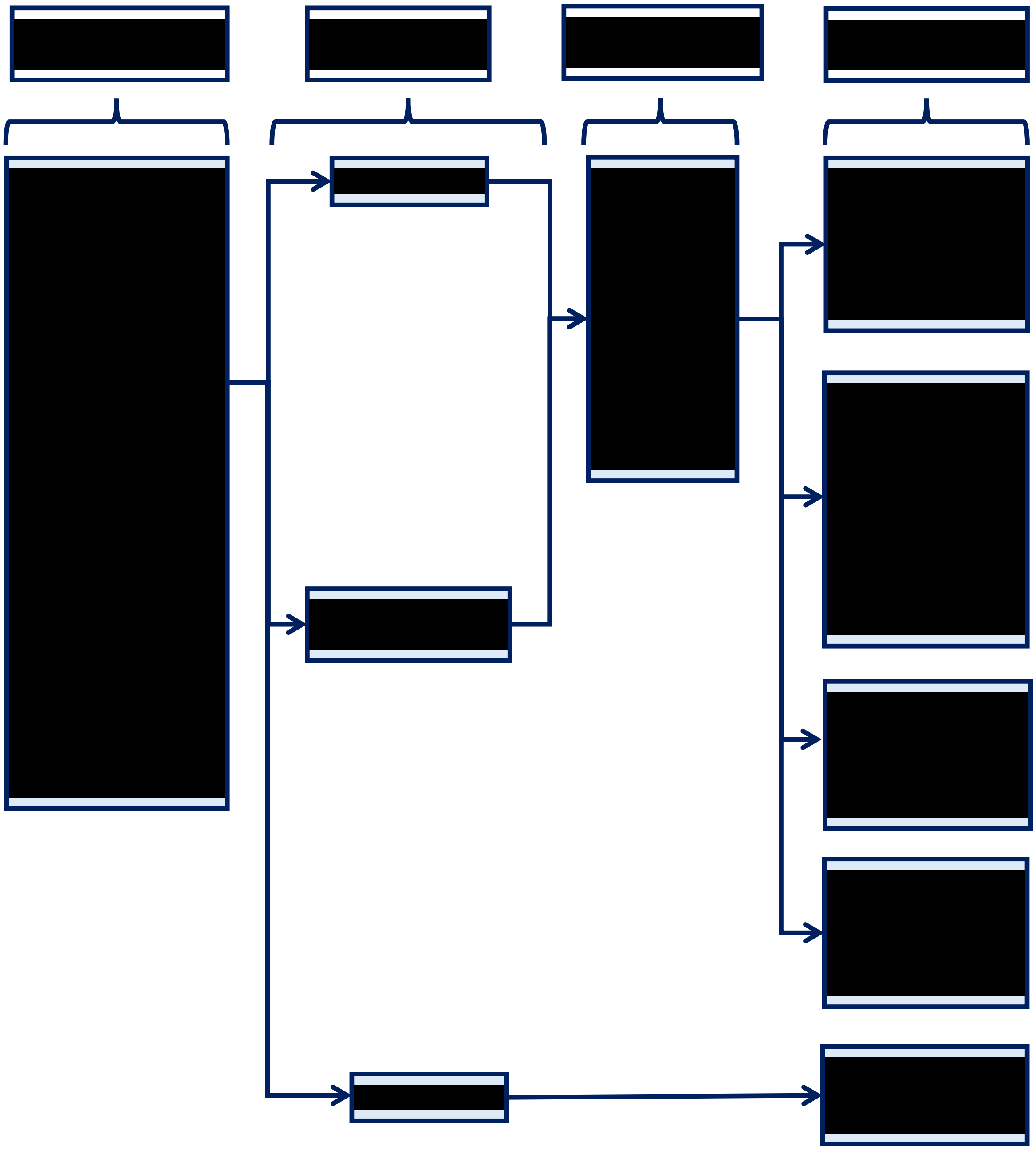


Figure 3.1 Flowchart showing the diversity of feedstock and the potential components involved in biorefineries based on the sugar or fermentation platform.[13-14]

Pretreatment uses both chemical, biological and mechanical processes to provide access to the insoluble polysaccharide component of lignocellulose for its subsequent hydrolysis into soluble hemicellulose (pentose: five carbon) and cellulose (hexose: six-carbon), and oligomers). The sugars are used as an intermediate platform to produce bioproducts and building block chemicals and materials alcohols, aldehydes, organic acids, amino acids, polyols. The typical fermentation method used for biofuels production is submerged cultivation (SmC). Alternatively solid substrate cultivation (SSC) is used, the growth of microorganisms on solid material in the absence of a free water process.[72] These fermentation processes could be batch, fed-batch, continuous or a combination of these fermentation schemes with integrated product recovery techniques such as gas stripping, pervaporation, extraction, adsorption, and pervaporation. Product concentration and purification uses the conventional distillation, adsorption, gas stripping, extraction, perstraction, reverse osmosis (RO) and pervaporation.[38, 64]

3.3.2 Process Systems Engineering

Process Systems Engineering is broadly defined as "analysis, design, optimization, operation, and control of complex process systems as well as the development of model-based methods and tools that allow systematic development of processes and products across a wide range of systems involving physical and chemical change"[20] and/or biological processing operation.[21] PSE encompasses both an academic and technological field that develop methodologies that are used as a decision-support for chemical engineering problems, spanning the supply chain creation and operation from the discovery, design, manufacture, and distribution processes.[73] The development of methodologies is responsible for planning, designing, operating, and controlling different unit operations, a chemical or production process, and an entire industry at large from a backbone of mathematical models and systems engineering tools.[73] This allows deeper understanding based on the development of systematic processes for small systems to large-scale batch and continuous industrial processes.[74] PSE offers tremendous avenues for evaluating process options, process integration, performance of building blocks and chemicals, biorefinery, and biocatalyst design.[75] To reap the full benefits of PSE tools and methods applied to fermentation-based processes as the chemical and refinery processes have enjoyed, the application of PSE tools fermentation-based biorefineries should move from unit-based to a holistic approach of the entire system with sustainability as the bedrock.[20]

3.4 The role and challenges of PSE in process design, simulation, and optimization

One of the principle benefits of using *in silico* platforms is the ability to consider large numbers of variables (process inputs and conditions) in optimization processes. If one thinks of a process simulation as a virtual laboratory to be used for experimentation, it becomes clear that considering large numbers of process options is much more feasible. Additionally, simulation allows the design engineer to add other downstream unit operations – operations that would not normally be included in traditional laboratory settings. This allows global optimization studies to be conducted where potential profitability is the objective. Insights as to how recycle options, upstream pretreatment and downstream separation and purification processes influence profitability can easily be gained. Apply PSE tools such as Monte Carlo simulation, Pareto Optimization, and Multi-Objective Optimization can only be practical *in silico*. Further, effective life cycle assessments and environmental impact assessments can only be conducted when a complete process model is in place. This underscores the importance of robust mathematical models. In short, having robust mathematical models is critical for developing an understanding of the process as a whole and what process variables have the greatest influence on profitability and environmental performance.

3.4.1 Challenges in using commercial and mature PSE tools for fermentation-based simulations

PSE tools are central to understanding and solving the sheer volume of decisions in fermentation-based processes arising from numerous process variables that must be optimized. Mature and commercial process simulators include the Aspen Engineering Suite™: e.g. Aspen Plus and Aspen HYSYS (Aspen Technology; Cambridge, Massachusetts),[29-31] gPROMS (Process Systems Enterprise; London, UK),[29-30] PRO/II (Schneider Electric Software, CA, USA),[29] UniSim® Design Suite (Honeywell International, Inc.; Morris Plains, NJ), Extend (Image That, San Jose, CA), and SuperPro Designer® (Intellingen; Scott's Plain, NJ).[30, 32] Most process simulators were traditionally designed for the petrochemical and refinery industries which are continuous processes, often operated in steady state mode apart from the transient processes (start-ups and shutdowns).[32] PSE tools and methods played a critical role in expanding the manufacturing capacity and developing economical viable processes in the chemical and refinery industries.[35]

However, PSE tools have not been readily applied to fermentation-based biorefineries because of lack of unit operation models with complete features that support the simulation of complex fermentation processes, (i.e. simulation of a fermentor as unsteady state batch process) and key process components, such as cells, in the databank.[2, 33] PSE tools such as SuperPro Designer® and Aspen Batch Process Developer (Aspen Technology; Cambridge, Massachusetts) dedicated to the simulation of batch bioprocesses[32] also lack some fermentation-based components in the database and the built-in rate-based kinetics generally cannot be used for mechanistic models (ordinary differential equations, ODEs) describing batch fermentations. Most fermentation processes are integrated with *in situ* product recovery processes to alleviate product toxicity to microorganisms. Traditional PSE tools such as Aspen Plus do not have in-built models that can readily simulate integrated fermentation and *in situ* product recovery processes.

3.4.1.1 Databases and physical property methods for bio-based components in traditional PSE tools

As shown in Figure 3.1, the diversity of the potential feedstock, variable composition of each feedstock (30 – 50% and 10 – 25% of dry weight for glucose and xylose contents in various feedstock, respectively) and sheer volume of components present a challenge in incorporating all these components in the databank of process simulators, providing the physical property data required and using PSE tools in the fermentation platform in a biorefinery. Although some efforts have been made to include typical components involved in fermentation processes in the databank of traditional process simulators, enzymes, yeast, and other microorganisms are still nondatabank components or missing physical property data when they exist in databanks in process simulators. This is because underdefined components, such as enzymes and cells, are biological species and usually have unknown structures, and are difficult to characterize. These result in missing physical property data in the databank of process simulators.[76] The successful simulation of fermentation-based biorefineries require the physical properties of components involved in the fermentation in the databank of process simulators for all component compositions, temperature, and pressure ranges. The validity and precision of process simulations depend on accurate physical property and thermodynamic models employed. Sometimes, estimation/prediction of many pure components and mixture properties are required because these properties cannot be measured or are difficult to measure experimentally,[77] requiring user-supplied data in process simulators.[78]

Eric C. Carlson[79] suggested a five-step process to describe the physical properties of components in a simulation: selecting the appropriate physical property method, validating the physical properties predicted by the selected method, simulating nondatabank components, obtaining, and using physical property data. Physical property estimation methods utilize property models that predict molecular properties from structural descriptors. Physical property models can be grouped into empirical, semiempirical and mechanical models, which together predict pure components, mixtures and polymer physical properties. Examples of physical property models include group contributions ("the properties of a molecule are determined by adding the occurrence frequency times the contribution of each molecular group in a molecule")[80] and topological indices ("the molecular descriptors are calculated using the interaction among different atoms/molecules groups, correlating the chemical structure to physical properties of a molecule").[80] Because of the advances in thermodynamics, an extensive collection of thermodynamic models (including equation-of-state, activity coefficient, predictive, and electrolyte based)[81] are available in PSE tools. The nature of the components, type of mixture, and range of conditions dictate the choice of thermodynamic models.[81]

3.4.2 Challenges in applying traditional PSE process simulators for optimization of fermentation-based biorefineries.

An optimization process finds the decision variables that represent and predict the optimum (maximum, minimum or target) of one or more objectives. Process optimizations are applied in many industries including chemical, oil and gas, refinery, pharmaceutical, and bioenergy, among other industries, to improve the process performance (generally increase profits and reduce costs), and minimize environmental and societal impacts of processes. Optimization of fermentation-based biorefineries, biochemical processes, uses mathematical models that represent and predict the forecast of the process given initial conditions and process inputs. This proposition is a familiar area that PSE plays a leading role. The desired performance of the process in the form of multiple objective functions are often inherently partially or fully competing[27-28] and require simultaneous objectives optimization subject to limited resources and process inputs (constraints – decision variables). Competing means as one parameter is enhanced (optimized), one or more parameters are simultaneously made worse or compromised. Examples of such competing pairs of objectives include product selectivity and conversion, product quality and recovery cost, profit and safety cost,[27-28] etc.

The traditional optimization approach has been the optimization of a single objective function, single objective optimization (SOO). In the case of a multiple objective system, SOO translates a multiple objective system into a single objective by either using relative weights to combine different objective functions or choosing one objective function as the main objective function while transforming other objectives into additional constraints. SOO finds a single unique solution unless multiple solutions exist where, local and global optima may be found by repeatedly solving the SOO problem with different initial guesses.[27-28] On the other hand, multiobjective optimization (MOO) finds a set of decision variables (Pareto domain) and their corresponding alternative sets of equally optimal solution(s) (Pareto-optimal solutions) by simultaneous optimization of (often competing) multiple objectives. Similar to SOO, MOO finds only one unique solution if the objectives are not competing. MOO often utilizes methods such as the Net Flow Method (NFM) and Rough Set Method (RSM) to analyze the Pareto-domain based on the interest of the decision-maker. Unlike the SOO that could hide the trade-offs that exist among competing objectives (process performance parameters), MOO provides insightful information about the trade-offs. Furthermore, MOO gives alternative solutions that can give guidance and insight about the performance of a process to aid in process design and control, and present what-if scenarios to mirror different process performances when process inputs and initial conditions change.[28]

Linearized and simplified or shortcut models are adequate for steady state and continuous chemical and refinery processes. Consequently, most traditional process simulators can only handle SOO (e.g. Aspen Plus) or have no built-in optimization tools (e.g. SuperPro Designer) as illustrated in Table 3.1 with two commercial process simulators. As shown in Table 3.1, Aspen Plus V9 can only handle SOO, using solvers such as the sequential quadratic programming (SQP) and complex method. To optimize fermentation processes linked to *in situ* product recovery techniques such as gas stripping, the formulation of the optimization problem requires process simulations based on time-dependent hydrolysis and fermentation (describing the inherent unsteady state process) models linked *in situ* to a separation process. MOO methods are required for fermentation-based processes because integrated fermentation and *in situ* separation processes often involve competing pairs of objectives such as selectivity and conversion, product quality and recovery cost, etc. MOO in traditional simulators will require linking the process simulator to an external software that can handle non-linear and MOO problems, such as MATLAB.

Table 3.1 Assessment of the databank and nondatabank components, and built-in rate based reaction kinetics and optimization tools available in Aspen Plus V9 and SuperPro Designer® v9.0.

Process simulator	Fermentation components in database	Built-in rate-based kinetics	Optimization tools available
Aspen Plus V9	*Databank*: glucose, xylose, sucrose, arabinose, galactose, mannose. *Nondatabank:* biomass, yeast, microorganisms.	*Power Law Model:* $r = kT^n \exp(-E/RT)\prod(C_i)^{\vartheta_i}$ *Langmuir-Hinshelwood Hougen-Watson (LHHW):* $r = \frac{(kinetic\ factor)(driving\ force)}{adsorption\ expression}$ *kinetic factor* = $kT^n \exp(-E/RT)$ $driving\ force = K_1\left(\prod C_j^{\vartheta_j}\right) - K_2\left(\prod C_j^{\vartheta_j}\right)$ $adsorption\ expression = \left\{\sum K_i\left(\prod C_j^{\vartheta_j}\right)\right\}^m$	*Type of optimization:* Only single objective *Solvers:* Sequential Quadratic Programming (SQP), Complex
SuperPro Designer® v9.0	*Databank:* glucose, xylose, sucrose, arabinose, galactose, mannose, biomass[a] and proteins[a] ($CH_{1.8}O_{0.5}N_{0.2}$) Corn steep liquor[a] (CHO) Molasses[b] and yeast[b] ($C_6H_{12}O_6$)	General rate expression (based on a reference component): $r_j = [\alpha\mu_{max}Term_1Term_2Term_3 + \beta](Term_4)$	No optimization tool is available.

[a] some physical property data are shared between biomass and protein components
[b] uses the physical property data of glucose

For Aspen Plus reaction kinetics: *r* is rate of reaction, *k* is pre-exponential factor, *T* is temperature, *n* is temperature exponent, *E* is activation energy, *R* is universal gas law constant, ϑ is concentration exponent, *i*, *j* are component index, π is the product operator, $\sum$ is the summation operator, *C* is component concentration, *m* is adsorption expression exponent, K_1, K_2, K_i are equilibrium constants.

SuperPro Designer® reaction kinetics: α and β are constants and μ_{max} is the maximum specific biomass growth rate. $Term_1$, $Term_2$ *and* $Term_3$ (can be used for inhibitor component) are kinetic expressions of three different components (e.g., substrates or reaction products). Monod, Haldane (with product inhibition), Inhibition, first order, and none are the options available that can be selected to represent $Term_1$, $Term_2$ and $Term_3$. $Term_4$ kinetic expression of the biomass component.

3.4.2.1 Examples of using traditional process simulators for optimization of fermentation-based biorefineries.

Due to the inherently nonlinear nature and complexity of optimization problems in fermentation-based processes, the available optimization tools in traditional process simulators are often inadequate. To put this in perspective, the classes of optimization problems encountered in biorefineries could be linear programming, quadratic programming, nonlinear programming, combinational optimization, dynamic optimization, mixed integer linear and non-linear programming, optimization under uncertainty, bilevel optimization, global optimization, and multiobjective optimization.[27] The determination of most objective function values and sometimes constraints require rigorous thermodynamic models in order for the optimization schemes to be realistic and practical. The following examples (not exhaustive) show how traditional process simulators have been linked with various external programs that have rigorous optimization tools for the optimization of fermentation-based biorefineries.

Gudena et al.[27] transformed a MOO problem using the ε–constraint method, where one objective from a multiple objective system was chosen as the main objective function while the other objectives were transformed into additional constraints for a hybrid steam stripper-membrane process for continuous bioethanol purification. The resulting SOO problem was solved repeatedly in Aspen Plus using several small changes in a chosen "ε" to obtain a set of optimal solutions. You et al.[82] linked Aspen Plus (determines data required to evaluate objective function values) to a multiobjective mixed integer linear programming in GAMS and used the ε-constraint method to find Pareto-optimal curves that showed the trade-off between economic, environmental, and social factors of a cellulosic ethanol (biofuel) supply chain.

Vázquez-Ojeda et al.[83] used a stochastic global optimization algorithm (differential evolution) implemented in MATLAB and coupled to rigorous process simulations in Aspen Plus for SOO of the purification of ethanol in a liquid–liquid extraction process (investigated extraction using conventional ethylene glycol versus octanoic acid, octanol and iso-octanol (ethylhexanol) as extractive distillation agents). The communication between MATLAB and Aspen Plus was via Microsoft Excel®. In another example, a multiobjective optimization of three integrated continuous ABE fermentation and *in situ* product recovery processes (gas stripping, pervaporation, and vacuum separation methods) was carried out using a Dual Population Evolutionary Algorithm (DPEA) to determine Pareto-optimal solutions and ranked using NFM to determine the best operating conditions of each separation technology. The Visual Basic Applications (VBA) program that solves the set of fermentation ODEs by the finite differences method was used as the simulation platform linked to Honeywell UniSim® software for the thermodynamic information.[84-85] Other interesting examples are found in the work of Geraili et al.[52, 69] who used an Aspen Plus process simulation (incorporate experimentally derived kinetics of complex biological reactions) linked to a hybrid strategic and operational level optimization framework in MATLAB through Aspen Plus ActiveX Automation technology to evaluate alternative technology options to select the optimal configuration from process yields and economic profit criteria.

3.4.3 Incorporation of rate kinetics in fermentation simulations using traditional process simulators.

Fermentations in biorefineries are frequently operated as batch or semibatch processes, in which the properties (concentrations, pH, temperature, etc.) change with time. As discussed previously, the form of kinetic models describing fermentation processes typically does not conform to the built-in reaction kinetic models even in dedicated batch process simulators, like SuperPro Designer®, as illustrated with two mature and commercial process simulators in Table 3.1. For example in Aspen Plus V9, only two built-in rate based kinetics exist: power law and Langmuir-Hinshelwood-Hougen-Watson. SuperPro Designer v9 on the other hand has one general rate-based kinetics that gives a user the options to select a Monod, Haldane, substrate inhibition, first order-based dependency on microorganism and none. The use of different enzymes and microorganisms, that typically have different metabolic pathways and mechanisms of substrate utilization, product formation, substrate and product inhibitions, makes it infeasible to have built-in kinetic models in process simulators that can handle every form of rate kinetics.

3.4.4 Traditional simulation approaches for fermentation process in biorefineries

Batch fermentations have been simulated using stoichiometric reactors in which product distribution is relative to a key component such as glucose or xylose with fixed product yields.[5, 86-89] The autocatalytic production of cells and maintenance are either ignored or also represented with stoichiometric equations at fixed yields relative to the formation of other products. This simulation approach decouples the interdependency of substrates and products on microorganism concentrations and production and does not incorporate time-dependent cell growth with product and substrate inhibitions. For integrated batch and *in situ* product recovery techniques used to alleviate product inhibition, the traditional simulation approach decouple the interdependent batch fermentation from the *in situ* product recovery. For example, the integrated batch Acetone-Butanol-Ethanol (ABE) and *in situ* gas stripping process is traditionally simulated with a stoichiometric reactor (based on stoichiometric equations and coefficients eqs 3.1 – 3.7) to represent the final batch fermentation broth and a flash unit to simulate the final broth composition, from the stoichiometric reactor, as the gas stripping process.[5, 86-87]

$C_6H_{12}O_6 \rightarrow C_4H_{10}O$ (butanol) + $2CO_2 + H_2O$ (3.1)

$C_6H_{12}O_6 + H_2O \rightarrow C_3H_6O$ (acetone) + $3CO_2 + 4H_2$ (3.2)

$C_6H_{12}O_6 \rightarrow 2C_2H_5O$ (ethanol) + $2CO_2 + H_2$ (3.3)

$C_6H_{12}O_6 \rightarrow C_4H_8O_2$ (butyric acid) + $2CO_2 + 2H_2$ (3.4)

$C_6H_{12}O_6 \rightarrow 3C_2H_4O_2$ (acetic acid) (3.5)

$C_6H_{12}O_6 + 6O_2 \rightarrow 6H_2O + 6CO_2$ (cell maintenance) (3.6)

$C_6H_{12}O_6 + 1.1429NH_3 \rightarrow 5.7143ZYMO$ (Cell biomass) + $0.2857CO_2 + 2.5714H_2O$ (Cell growth) (3.7)

Approaches to incorporate time-dependent fermentation models into traditional process simulators include linking the process simulator to an external program that can integrate the typical form (e.g., ODEs) of rate-kinetics such as MATLAB, Excel, or a user-defined kinetic subroutine. Geraili et al.[52, 69] used a dynamic link and data exchange between Aspen Plus ActiveX Automation technology and MATLAB to incorporate kinetic models (ODEs) that describe enzymatic hydrolysis, and a Monod-type of cell growth to simulate batch fermentation for the production of ethanol and succinic acid. Similarly, Quintero and Cardona[90] simulated a batch fermentation for the production of ethanol from rice hulls in Aspen Plus considering the pretreatment, hydrolysis, detoxification and fermentation steps using a recombinant bacteria *Zymomonas mobilis*. The dilute acid and liquid hot water pretreatments, hemicellulose and cellulose hydrolysis were simulated with batch reactors with first order rate-based kinetics. The kinetic models describing the detoxification and batch fermentation were simulated using a user subroutine written in an Excel-MATLAB interface linked to Aspen Plus because these kinetic models did not conform to the built-in kinetic models in Aspen Plus. In another example, Dias et al.[91] simulated a batch production of bioethanol from sugarcane by fermentation using SuperPro Designer® linked to an Excel spreadsheet. The nondatabank components (sugar cane bagasse and sugar cane impurities) were simulated as user-defined components in SuperPro Designer.

3.4.5 Current state of PSE for fermentation process modeling

Mathematical models play a crucial role in understating and optimizing cellular kinetics and for that matter, bioprocesses.[33-35] The accuracy of the prediction of a simulation depends, in part, on the quality of the mathematical model employed, given process inputs and conditions.[36] Generally, mathematical models can be classified as stoichiometric, kinetic, or a hybrid of stoichiometric and kinetic models, and these models are used to describe cell and fermentation processes.[92] Two classes of models are generally used in simulations. First, empirical – representation of a system with an input-output relationship, in which details of the underlying phenomena and mechanism are not incorporated. The predictions of empirical models are often only accurate within the range of conditions and data used to formulate the model (interpolative). Second, mechanistic models represent a system by incorporating the constituent parts and underlying mechanism to formulate mathematically the internal operations of a system. In this way, the forecast (extrapolation)[2] of a system can be predicted from a given set of initial conditions and process inputs (deterministic principles that are based on the knowledge of the process) with a better accuracy.[93] The typical constituents of a fermentation system include substrates, cell biomass concentrations, working volume, etc. whereas the underlying mechanism includes microbial growth kinetics, mixing, heat and mass balances and transfer processes and thermodynamic phenomena.[33, 93]

When ideal mixing can be assumed in fermentations, mechanistic models can be classified into four categories based on microbial cell populations as shown in Figure 3.2. Structured models represent cellular behavior by viewing a cell as made up of various chemical components, whereas unstructured models consider a cell as a single chemical component. On the other hand, segregated models consider the individual cells that make up a population of cells to be different with diverse characteristics whereas unsegregated models represent a population of cells with an average cellular behavior, where for example all cells are deemed to have the same characteristics. For bioproce,sses, the traditional models have generally used empirical models using the well-known Monod expressions for microbial growth kinetics[93] and metabolites rates with mechanistic models for the physical process (mixing, heat transfer in the fermentation environment, etc).[2, 93]

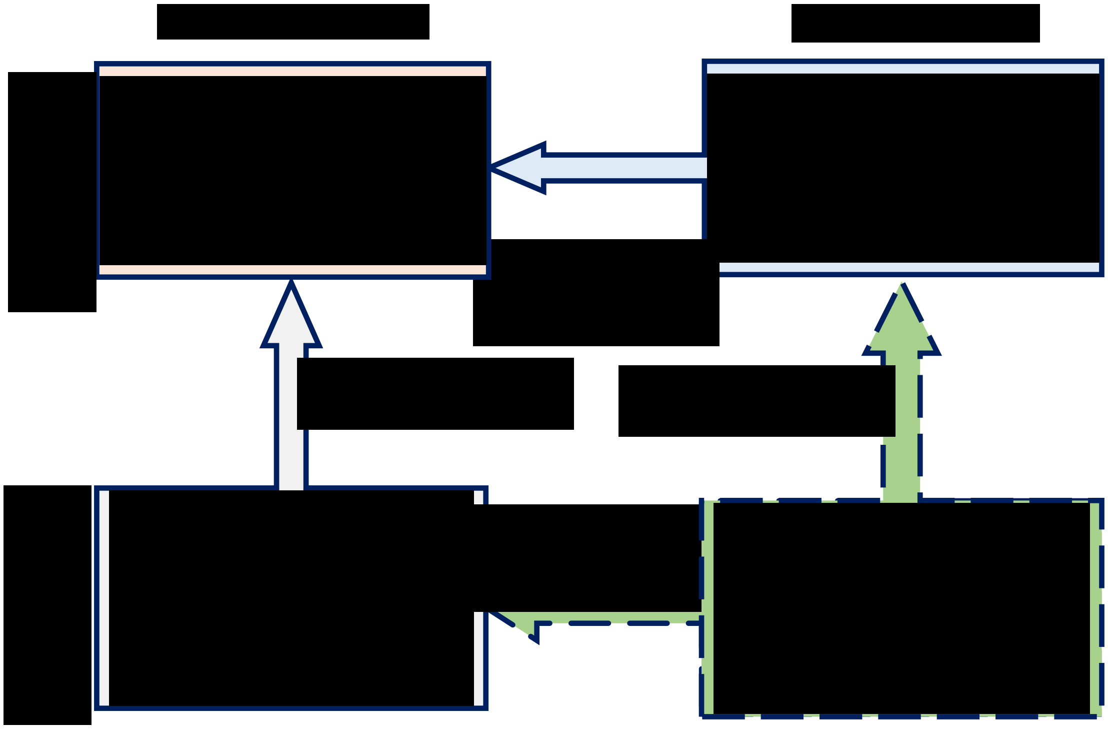


Figure 3.2 Mechanistic models classification based on cell population.[93]

Because of the complexity of bioprocesses, some of the underlying cellular mechanisms, such as regulatory mechanisms and stress responses, are not well understood to be described mathematically with mechanistic models, in which case the norm has been to use empirical models.[2] This challenge is exacerbated by the fact that the nonlinear behavior (time-dependency) of microorganisms determine the dynamics of bioprocesses, limiting detailed mathematical modeling because of partial microbial and fermentation kinetics knowledge.[35, 94]

3.4.5.1 The gap between developments in biology and mathematical models.

The advances in genome sequencing,[95-97] annotation tools and techniques,[97] multiomics data from experiments based on metabolites-protein interactions,[95, 98-99] DNA technology,[100] microchemical methods,[100] rational and evolutionary engineering strategies to improve the tolerance of microorganisms to fermentation, metabolic intermediates and end-product inhibitions[101] have been well documented and researched on an ongoing basis. Furthermore, advances and developments in process monitoring and control for bioprocesses have been impressive: real-time/near real-time measurements using infrared spectroscopy, Raman spectroscopy, capacitance sensors and mass spectroscopy for online real-time measurements of components concentrations (e.g., glucose, fructose, etc.), measurement of concentrations (e.g., glucose, acetate, formate, etc.), cell biomass (change of microbial morphology and viability) and real-time measurements of gases (CO_2 and O_2), respectively.[34] Nevertheless, mathematical model development and the use of computational tools have failed to keep up with these advances and developments in biology to further advance the knowledge in bioprocess understanding of fundamental biological concepts; most developed models rely on mathematical concepts and equations developed over a century ago (using Monod and Michaelis-Menten expressions).[35, 100]

In the case of fermentation-based processes, the biorefinery industry has been slow in translating and implementing the efforts of biochemical engineers (advances in metabolic and genetic engineering) into mathematical models.[35, 100] For example, *Saccharomyces cerevisiae* consume low concentrations of glucose and mainly respire to produce cell biomass and CO_2 whereas at glucose concentrations greater than 5 g/L, the microorganism undergoes an aerobic fermentation to produce cell biomass, ethanol and relatively higher CO_2 per glucose consumed.[100] Another example is the ABE producing *Clostridia* that produces ABE during the conventional batch fermentation using a carbon source through two separate growth phases: exponential acidogenic phase (butyric acid, acetic acid, carbon dioxide and hydrogen are produced) and late exponential and stationary solventogenic phase[4, 38] (excreted acids are taken up and converted to acetone and butanol in a typical ratio of 2:1).[4]

It must be mentioned that various researchers have investigated the switch from acidogenesis to solventogenesis in the ABE fermentation to understand the underlying mechanism using a systems biology approach.[102-104] There are also a hybrid of mechanistic models (for substrate uptake and utilization, formation of intermediate metabolites and solvent and cell biomass production) and mainly empirical models (describing regulatory mechanisms: on-off switch mechanism to cease metabolic reactions in the absence of substrate[105], pH-induced switch from acidogenesis to solventogenesis[106-108], dissociated and undissociated acid utilization[106-107, 109] and other regulatory mechanisms[105-109]). These two examples above illustrate how difficult it is to incorporate and translate the understanding of fermentation processes from experiments into mathematical models using understanding of the cellular machinery, regulatory, and stress response mechanisms to capture such trends.

3.5 A proposed integrated platform of PSE tools for time-dependent fermentation processes

As previously stated, commercial PSE tools, such as Aspen Plus, have been routinely and extensively applied to the design, analyses, and optimization of petroleum refinery processes.[73] However, these PSE tools lack unit operation models with complete features that support the simulation of fermentation processes, (i.e., reactors that can simulate unsteady state batch process), key process components, such as cells, in the databank[76] and robust optimization solvers that can solve the inherently nonlinear and multiple objective problem a biorefinery presents. Consequently, commercial PSE tools have found limited application to fermentation-based biorefineries. Process inputs in industrial batch fermentation processes are rarely constant (not steady state), and feedstock variability and various potential recycle streams from pretreatment, hydrolysis, fermentation, and purification steps[70] continuously alter process parameters in the fermentation process. Therefore, steady state simulations with fixed process yields are inadequate representations of the inherently unsteady state batch fermentation especially when batch fermentations are coupled with *in situ* product recovery techniques, such as gas stripping. Consequently, steady state simulations of integrated batch fermentation and *in situ* product recovery cannot be used for process optimization if the goal is to optimize the operating conditions.

Three major requirements needed to utilize existing PSE tools for the design, analyses, and optimization of fermentation process are the simulation of nondatabank components, simulation of batch fermentations based on time-dependent fermentation models, and linking traditional PSE simulators to robust and nonlinear optimization solvers. A general proposed framework that links a robust non-linear optimization solver to a traditional process simulator that has the ability to simulate an unsteady state batch fermentation process is shown in Figure 3.3. A communication platform that links the process simulator to the optimization solver is required, allowing a two-way communication between the process simulator and the optimization software. In this way, data can be written and read between the two platforms. Based on the decision variables and constraints (indicating the range of each decision variable), different combinations of the process inputs and operating conditions can be communicated from optimization solver to process simulator and the corresponding data required to evaluate objective function values communicated from process simulator to optimization solvers through the communication platform. This establishes a circle of communication from the optimization software to the process simulator through the communication platform.

The optimization process requires the determination of objective function values subject to constraints (*CN*). The objective function values must be determined for each set of decision variables. The decision variables are passed from the optimization platform to the process simulator through the communication platform. In the process simulation platform, the fermentation process is simulated as unsteady state process using a time-dependent fermentation model, where the thermodynamic models in the process simulator offer opportunities to integrate fermentations with *in situ* separations. The data required to evaluate the objective function values in the optimization platform are read from the process simulator through the communication platform. Depending on the optimization algorithm used, an improved set of decision variables may be generated for further iterations until the stopping criteria is met.

The built-in generalized reaction kinetics in commercial simulators cannot be used for the ODEs describing batch fermentation processes because the ODEs do not conform to the built-in generalized kinetics as illustrated in Table 3.1. To incorporate ODEs into the kinetics of batch unit operations, the batch unit has to be customized, where the process simulator is linked to a user defined subroutine or an external program that solves ODEs, such as MATLAB or Aspen Custom Modeler. To simulate non-databank components such as cells, component substitution can be used. Component substitution is the use of the known physical property data for the unknown properties of a nondatabank components. Microorganisms (e.g. cells) can be simulated as solids with the physical property of another component. As solids, the microorganisms will not participate in the vapor-liquid equilibrium and interfere with the thermodynamic calculations.[110]

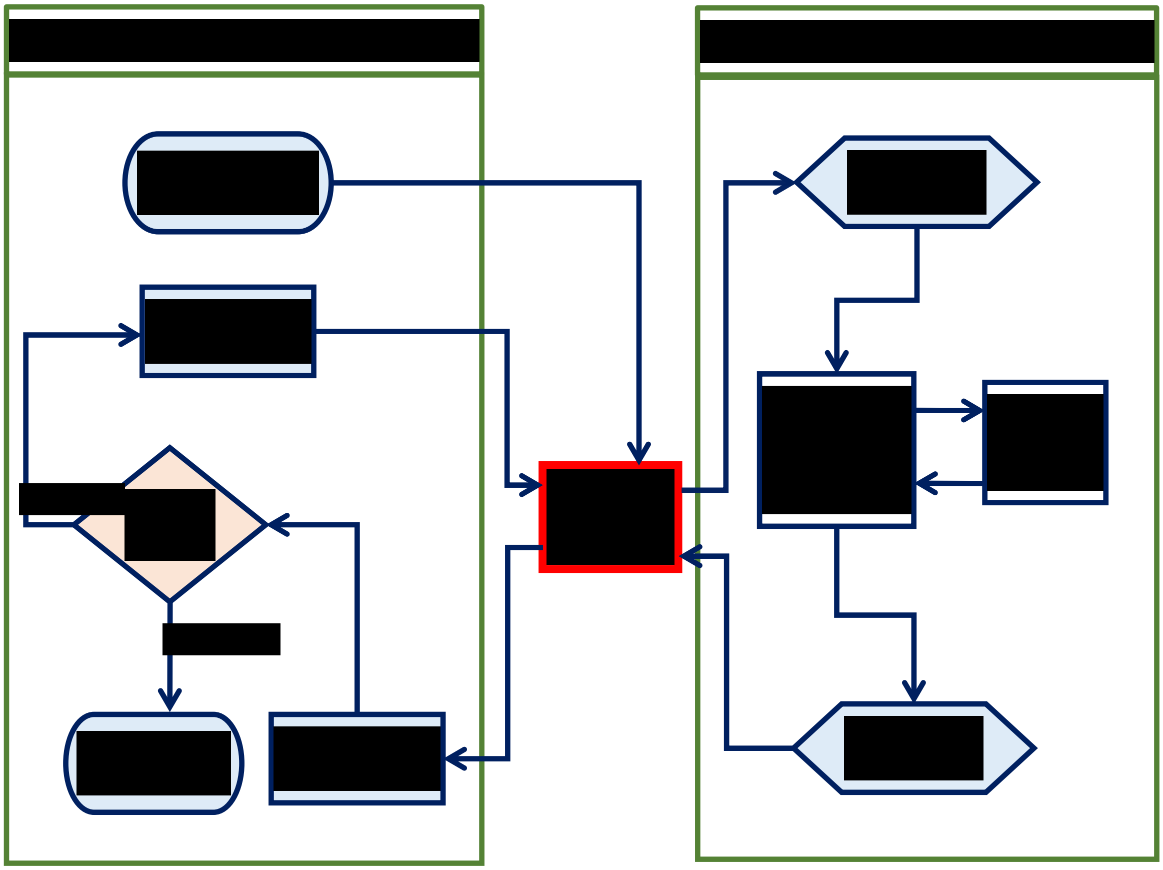


Figure 3.3 General framework for process simulation and optimization of fermentation processes based on a time-dependent (kinetic) model.
The notation is decision variables (*DV*), objective functions (*OF*), constraints (*CN*), and stopping criteria (*SC*).

This framework can be applied to any fermentation-based problem when the following requirements are met: (1) A traditional process simulator has been customized to simulate fermentation as an unsteady state process (for example, using a user defined subroutine or an external program that can solve ODEs). (2) Non-databank components are simulated. (3) A communication platform is identified (e.g., Visual Basic for Applications

(VBA) in Microsoft® Excel or component object model (COM) interface in MATLAB) to link the traditional process simulator to a rigorous optimization platform.

3.6 Conclusions and future directions

The current state-of-the-art PSE tools in bioprocess simulation and modeling still have room for improvement and innovation. In order to improve titers, yields, and productivities in fermentation-based processes, the cellular machinery (based on gene functions and enzyme kinetics, regulatory network, signal network and stress response mechanisms of cellular kinetics,[92, 111] and development of advanced genome editing tools[111]) and the extracellular fermentation environment (containing nutrients, substrates and other components necessary for metabolism) must be optimized. In-depth understanding of microbial kinetics and bioreactor dynamics is imperative to the optimization of bioprocesses. A combined model framework of cellular kinetics and bioreactor dynamics (interdependent factors) can be used in the design and optimization of fermentation processes.[34, 92] In the interim, an integration of stoichiometric with dynamic models, regulatory, and signaling mechanism models taking into account uncertainty[13] could guide cell engineering and bioprocesses based on lignocellulosic biomass.[92]

Cellular kinetics require time-dependent models (can be used to improve substrate utilization, product titers, and yields) and cellular engineering (used to improve the bioprocess performance based on metabolic engineering, genetic engineering, and understanding of cellular regulation mechanism and the resulting improved process design from mathematical models) to overcome theoretical performance limitations of in using microorganisms.[34] Mathematical models should move away from steady state and stoichiometry to mechanistic (kinetic) models that describe cellular metabolism and regulation mechanisms, driven by extracellular changes/perturbations and genetic modifications.[34] These models must be tailored to lignocellulosic biomass-based systems.[13] Although unsegregated unstructured models (top left corner, Figure 3.2) are the commonly used for bioprocesses (for the overall microbial and fermentation kinetics),[92-93] these models should be replaced with segregated structured models (bottom right corner, Figure 3.2) that are able to describe the underlying cellular mechanisms (intracellular, intercellular, and extracellular, considering variables such as nicotinamide adenine dinucleotide (NADH), adenosine triphosphate (ATP), intermediate metabolites and cell biomass), metabolic flux analysis (MFA), flux balance analysis (FBA), and genome-scale models.[2, 13, 35, 93] It is important to incorporate the kinetics of inhibitors (substrate, products, inhibitory compounds produced from the pretreatment and hydrolysis steps) in the cell growth kinetic models.[92]

The pretreatment, chemical and enzymatic hydrolysis processes should also be based on a combined model-based approach and experiments.[92] Because of the sheer number of microorganisms and enzymes used in fermentation processes, mathematical models developed for one system will generally lack applicability to other processes without a complete reestimation and evaluation of the model parameters.[35] An adaptive parameter system that accounts for the observed variations in parameters used in mechanistic models should replace the use of constant model parameters for the yields and process rates (usually change over time). In this case, advanced monitoring and control systems can be used in bioprocesses to offer insights and trends in these parameters, reporting the sensitivity analyses of parameters and confidence intervals with models developed.[2, 95] A holistic framework that integrates model-based methods and tools with experiments on a multidisciplinary platform is needed to utilize the expertise from biochemical engineering, bioengineering, molecular science (innovation and understanding the cellular mechanisms), and process system engineering (exploratory research guidance)[35, 112] to explore the operational space using process simulations.[2] It must be noted that significant investments in terms of resources, time, and in-depth process insight are required to develop mechanistic models.[2] The challenge here is developing models that are complex (incorporate time-dependency) and robust to predict bioprocess dynamics while remaining computationally tractable.[35, 93, 113] The ultimate goal in developing these dynamic segregated structured models should be to translate the process understanding and findings to simplified models that can readily be used in industrial fermentations.[93] Comprehensive reviews on mathematical model developments can be found in the work of Koutinas et al.[35] and Motta and Pappalardo.[114]

Process simulators need adaptation in order to be readily applicable to fermentation-based biorefineries. Because of the inherent nonlinear nature of microbial and fermentation kinetics, existing process simulators need to add unit operation models or modify existing unit operation models, such as batch reactors, to be able to handle time-dependency and offer unsteady state batch simulations. In the interim, developers of traditional process simulators should explore avenues to expand the capabilities of existing simulators and offer tutorials and examples on how to link these simulators to other PSE tools that have functionalities and features needed to simulate fermentation processes. Without robust simulation models that reflect the intricacies of fermentation processes, it is not possible to employ other tools such as SOO or MOO to refine the process design and operating parameters. Furthermore, there is a need for the development of analytical tools that can characterize substrate (lignocellulosic biomass) structural properties, correlated with pretreatment and hydrolysis, and microorganism physical properties.[13] In this way, the databanks in a traditional process simulator can be updated to include the typical lignocellulosic-to-bioproducts components with the required physical property data.

Fermentation processes often utilize *in situ* product separation to remove products, as it is produced, to prevent the build-up of products to inhibitory levels. In traditional process simulators, mature separation processes such as distillation and extraction can be readily simulated. However, commercial and mature process simulators such as Aspen Plus generally lack the capability to simulate *in situ* product recovery techniques[13] such as gas stripping, membrane separation, pervaporation, and adsorption that have been experimentally applied to alleviate product inhibition to microorganism in fermentations. There is an urgent need to expand the capabilities of unit operation models in traditional process simulators to accommodate these separation processes. The virtual experiments and process analysis using process simulators saves time and resources by indicating the most impactful experiments to conduct for process design, continuous improvements, and optimization,[32, 36] providing sustainable processes and products. When these tools become commercially available, biomass processes may at last prove to be economically viable and true sustainable alternatives to traditional, petrochemical products.

Chapter 4

Unsteady State Process Simulation of Integrated Batch Acetone-Butanol-Ethanol (ABE) Fermentation and *in situ* Gas Stripping

4.1 Summary

Process simulations of batch fermentations with *in situ* separation traditionally decouple these interdependent steps by simulating a “steady state” fermentation broth representative of the final batch fermentation broth composition and uses this stream as an input to a steady state separation unit. In the case of integrated batch fermentation and *in situ* gas stripping, this approach may be inadequate for the systematic analyses of the process because the fermentation kinetics are directly linked to the gas stripping process. In this study, batch acetone-butanol-ethanol (ABE) fermentation is used as a model system to translate time-dependent fermentation models to Aspen Plus using a FORTRAN user kinetics subroutine linked to the batch reactor, an approach validated using MATLAB. Consistent with literature experimental results, batch fermentation and *in situ* gas stripping simulations predict an improvement of the total ABE produced, yield and productivity compared with the batch process, demonstrating the ability of the thermodynamic models to predict phase concentrations/compositions in fermentation and *in situ* separations process simulations. The ABE product profile is sensitive to the gas flow rate, unlike previous separate steady state fermentation and equilibrium-based gas stripping simulations, demonstrating the importance of a linked fermentation and separation simulation approach for the systematic analyses of the process. A batch fermentor simulated using a time-dependent fermentation model linked directly to separations unit operations can provide strategies that can serve as a decision-support tool to the fermentation experimentalist and bioprocess design engineer.

4.2 Introduction

The production of biofuels and chemicals from lignocellulosic biomass via fermentation platform is considered a sustainable energy alternative to fossil fuels.[1] The alcohol products of the acetone-butanol-ethanol (ABE) fermentation process, butanol and ethanol, have properties that make them excellent liquid transportation fuels.[37-38] Additionally, ABE products are used as solvents and for the production of other chemicals.[43] The ABE fermentation is characterized by low final ABE concentrations (1

– 2 wt. % butanol[38] or less than 20 g/L ABE[11]), low ABE yield (0.28 – 0.33g/g) and low reactor productivities (less than 0.3 g/L/h) as a result of butanol toxicity to the microorganisms.[11] Approaches to reduce product toxicity and enhance the productivity and yield of the ABE fermentation process include integrated batch fermentation and *in situ* product recovery techniques, such as gas stripping, adsorption, pervaporation, liquid-liquid extraction, perstraction, and reverse osmosis.[1, 38, 42] Gas stripping as a chemical separation method allows for the selective removal of the volatile components (ABE) from the aqueous fermentation broth by continuously bubbling a gas through the fermentation broth, preventing the buildup of butanol to inhibitory concentrations.[38, 42, 115] Gas stripping is relatively simple, has the option of using the fermentation product gases (carbon dioxide and hydrogen) or another inert gas as the stripping gas and can be operated at the fermentation temperature.[7]

Process Systems Engineering (PSE), which employs a sophisticated toolset to simulate a mathematical representation of a process, played a critical role in understanding and expanding the production capacity of the chemical and refinery industries.[35] Examples of PSE commercial process simulators include Aspen Engineering Suite™ (Aspen Technology; Cambridge, Massachusetts) and gPROMS (PSE; London, UK), UniSim® Design Suite (Honeywell International, Inc.; Morris Plains, NJ), Extend (Image That, San Jose, CA), and SuperPro Designer® (Intellingen; Scott's Plain, NJ). Aspen Plus, the most commonly used product in Aspen Engineering Suite™ (AES), is a universally accepted commercial steady state process simulator.[116] Simulations can use shortcut (linear/stoichiometric equations), rigorous (relying on time-dependent/kinetic or detailed mechanistic models), and a hybrid of shortcut and rigorous methods. Generally, rigorous method-based simulations have greater predictive capabilities compared with shortcut-based simulations.[117]

In a fermentation process, microorganisms extract nutrients and convert them into more cells, energy and metabolic products within a controlled environment (autocatalytic production of cells; Substrate(s) + Cells → Extracellular Products + More Cells).[6] In an integrated fermentation and *in situ* product recovery system, the fermentation kinetics and separations are coupled on a time-dependent basis. Steady state simulations, which decouple the time-dependence of the fermentation and separation unit operations, have traditionally been used to simulate the batch ABE fermentation and *in situ* product recovery by gas stripping.[5, 86-87] (Figure 4.1A). In the steady state simulation, the ABE fermentation is described using a steady-state reactor that uses stoichiometric equations with fixed product yields and distributions relative to a key feed component (e.g. glucose or xylose).[5, 86-87] The autocatalytic production of cells in previous simulations were either ignored[87, 118-119] or represented with stoichiometric equations in which cell maintenance or growth was at a fixed yield relative to the formation of other products,[5] thus removing the time-dependent cell growth, substrate consumption, and product inhibition.

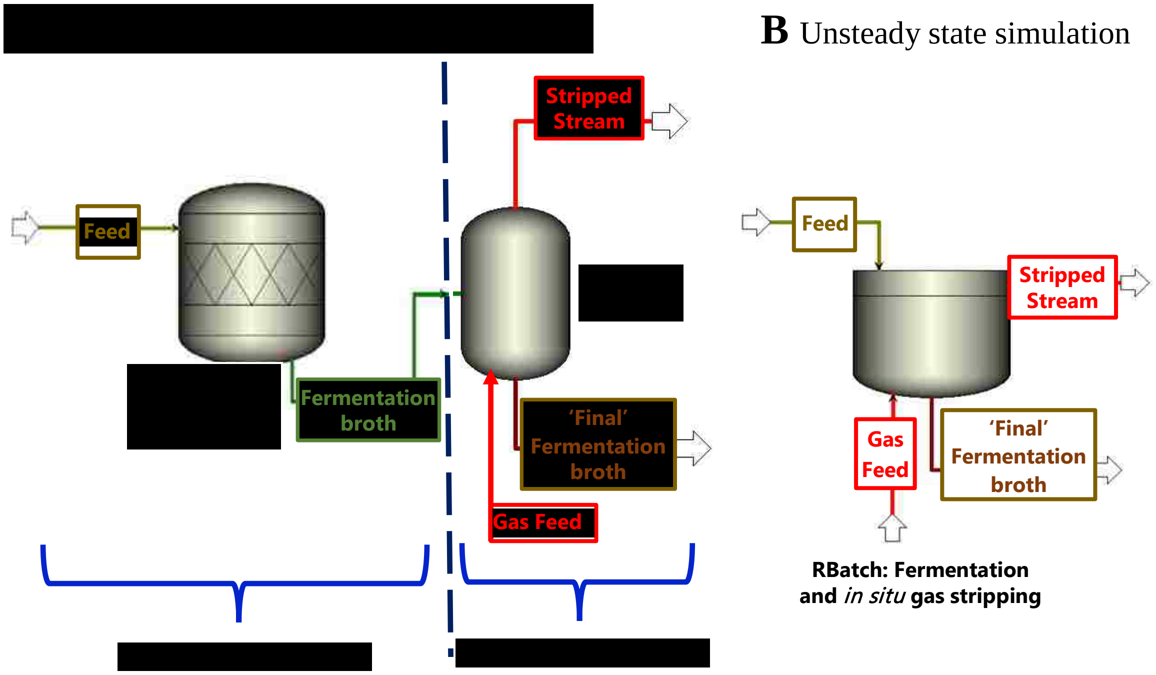

[redacted]
B Unsteady state simulation
Stripped Stream
Feed
[redacted]
[redacted]
Fermentation broth
'Final' Fermentation broth
Gas Feed
[redacted]
[redacted]
Feed
Stripped Stream
Gas Feed
'Final' Fermentation broth
RBatch: Fermentation and *in situ* gas stripping

Figure 4.1 Simulation of integrated batch ABE fermentation and *in situ* gas stripping A) as separate steady state fermentation and equilibrium-based gas stripping unit operations in Aspen Plus; B) as unsteady state batch fermentation coupled with *in situ* gas stripping using a time-dependent fermentation model to link the fermentation kinetics with gas stripping (separations) in Aspen Plus

In traditional steady state analysis, the final fermentation broth from the stoichiometric reactor is then simulated with a flash unit (an equilibrium-based separator) to represent the gas stripping process. This simulation approach results in steady state simulations that decouple the ABE fermentation kinetics from the fermentation environment (which is a function of the gas stripping process). Batch and semi-batch fermentation kinetics change with time in response to the fermentation environment because of variable substrate availability, time-dependent concentration of cell biomass, and products. Thus, the separate steady state fermentation and gas stripping simulations may not be representative of the integrated batch ABE fermentation and *in situ* gas stripping. In addition, simulations based on steady state, stoichiometric, reactors may be inadequate for the systematic analyses of time-dependent processes of batch fermentations directly coupled to other unit operations, such as separation processes.

As an alternative to traditional steady state simulations, this work demonstrates an approach for the unsteady state simulation of the batch ABE fermentation and *in situ* gas stripping, allowing the operating conditions of the fermentation process, such as the gas flow rate, to be directly linked to the fermentation kinetics (Figure 4.1B). A cell-based kinetics mathematical model, a system of ordinary differential equations (ODEs) describing the ABE fermentation developed by Votruba et al.,[120] is used as a model system to develop techniques in Aspen Plus to simulate an unsteady state batch fermentation. The batch reactor in Aspen Plus, RBatch block, is linked to a FORTRAN user kinetics subroutine (calculating the rates of generation or consumption of each component) with a gas continuously fed to the reactor to simulate the unsteady state batch fermentation and *in situ* gas stripping process. The results of the integration of the system of ODEs describing the ABE fermentation process in MATLAB are compared with the batch simulation results in Aspen Plus (in the absence of gas stripping) to verify the accuracy of the developed procedure in Aspen Plus. The simulation results are compared to experimental trends observed in the available literature for ABE batch fermentation and *in situ* gas stripping as a function of gas flow rates. Furthermore, integrated batch fermentation and *in situ* gas stripping simulations are compared with traditional equilibrium-based steady state simulations of a steady-state fermentor with gas stripping of the final fermentation broth.

4.3 Methods

4.3.1 Prerequisites for batch fermentation simulation in Aspen Plus

4.3.1.1 Fermentation model used in the simulation

The efficient design, scale-up and optimization of cellular-based processes, such as the ABE fermentation, rely on mathematical model-based simulations, which offer insight at both the micro- and macro-scale of the process. A comprehensive review by Mayank et al.[121] compiled the mathematical models of the ABE fermentation process, which range from simple stoichiometric fermentation equations to more complex metabolic pathway-based kinetic models incorporating the autocatalytic production of cells, the dynamic changes in metabolites concentrations, and substrate and product inhibitions. The kinetic mathematical model developed by Votruba et al.,[120] which is based on the metabolic pathway for a batch culture of *Clostridium acetobutylicum*, was selected to simulate the ABE fermentation process using typical fermentation initial conditions of 50 g/L glucose and 0.03 g/L cell biomass. The model is based upon formulated mass balances from experimental data and trends in the form of rate equations for substrate consumption, the production of measurable extracellular products (acetone, butanol, ethanol, acetic acid, butyric acid, carbon dioxide, and hydrogen), the autocatalytic production of cell biomass, and product and substrate inhibition of cell growth. The mathematical representation of the fermentation kinetics and parameters of the model are presented in Appendix A and B, respectively.

4.3.1.2 Non-databank components in the simulation

Cells and some fermentation intermediate products are not found in the databank of Aspen Plus because of unknown physical properties or structures of these components. Component substitution, in which all the known physical properties of another component are used for all the unknown physical parameters of the non-databank component, is a simple and powerful technique to simulate non-databank components. The non-databank components in the fermentation kinetics model, cell biomass (C_X in eq A.2) and the physiological marker (PM in eq A.1) in Appendix A, were simulated as user-defined solid components with all the known physical properties of water from Aspen Plus databank. Component substitution is applicable to these non-databank components because they are not major components in the fermentation mixture. Furthermore, the non-databank components are non-volatile, non-polar, and do not participate in the vapor-liquid equilibrium (VLE) calculations as solids.[79]

4.3.1.3 Thermodynamic models in the simulation

The ABE fermentation mixture was modeled as a mixed aqueous and organic stream, with solid (due to the non-databank components), liquid and vapor phases. The nonrandom two-liquid – Hayden O’Connell (NRTL-HOC) property model was selected as the thermodynamic model for the simulation.[115] The NRTL activity coefficient model was selected to account for the nonideality of the liquid mixture as a function of temperature and composition.[79, 81] The fermentation system contains carboxylic acids, butyric and acetic acids, which form a strong association in the vapor phase. The HOC equation of state calculates the thermodynamic properties of these acids in the vapor phase by incorporating the chemical theory of dimerization. The fermentation mixture also contains the light gases, carbon dioxide (CO_2) and hydrogen (H_2), and nitrogen gas (N_2) used in gas stripping at concentrations less than 5 %, at a temperature above the critical temperatures of the pure components (CO_2, H_2, and N_2) and in subcritical solvents.[79] These components were therefore declared as Henry's components in Aspen Plus to account for dissolved product gas components in liquid fermentation mixture.[115]

4.3.2 Simulation procedure for batch fermentation using Aspen Plus

Aspen Plus is a steady state process simulator; the concentration, composition and other properties of a simulated process do not change with time. In contrast, the batch fermentation is an unsteady state process and the concentration of components and other properties changes with time. The simulation of an integrated batch and *in situ* gas stripping process requires a time-dependent fermentation model to be coupled with a stream of gas continuously bubbled through the fermentation broth to remove the volatile components (ABE) selectively as they are produced from the fermentation broth, preventing the buildup of ABE products to inhibitory levels. To simulate the batch fermentation as an unsteady state process in Aspen Plus, a time-dependent or kinetic model can be built in either Aspen Custom Modeler (ACM) or Aspen Plus Dynamics (APD) and exported to Aspen Plus. Non-standard kinetic models exported from ACM or APD to Aspen Plus do not have the option to accept additional feed/input streams, such as a continuous feed of a gas, to simulate gas stripping. The batch reactor, RBatch unit operation, in Aspen Plus is able to simulate unsteady state batch and semi-batch processes rigorously. The RBatch uses holding tanks to interface the steady state flowsheet environment in Aspen Plus and the unsteady state batch operation. The RBatch block allows for a dynamic continuous feed, which is a steady state flowsheet stream fed continuously during the batch operation. These two features of the RBatch block present a unique opportunity for the unsteady state simulation of the batch fermentation alone or with *in situ* product recovery techniques, such as gas stripping. The unsteady state simulation of a batch fermentation process in Aspen

Plus requires the conversion of the batch charge (50 g/L glucose and 0.03 g/L cell biomass) to steady state flow rates (and vice versa) and linking the batch reactor to a FORTRAN user kinetics subroutine that calculates the rates of consumption/production of each component in the reactor through the ODEs describing the fermentation process as summarized in Figure 4.2.

4.3.2.1 Interfacing the continuous steady state flowsheet environment in Aspen Plus with the unsteady state batch reactor

To interface the steady state process simulation in Aspen Plus and the inherently unsteady state batch reactor, the RBatch block uses holding tanks to accumulate material from a feed, vent and product streams as shown in Figure 4.3. The accumulated material in the holding tanks can then be used to convert material streams from the steady state environment in Aspen Plus to a time-dependent system in the inherently unsteady state batch reactor as time-averaged streams and vice versa. A specified reactor cycle time or batch feed time can be used to convert data between the continuous steady state and time-varying batch operation. The total cycle time is specified for the batch reactor, ensuring that the mass balance between the RBatch reactor input and output streams is achieved. The initial batch charge of 50 g/L glucose, 0.03 g/L cell biomass in a 1 L aqueous solution were converted into steady state feed flow rates using a total cycle time of 1 minute. The resulting feed to the RBatch block was specified on a total volumetric flow basis of 1 L/min with a mass concentration of: glucose (50 g/L), cell biomass (0.03 g/L), the physiological marker, *PM*, (1 g/L) and water specified as the solvent to form an aqueous mixture with the specified concentrations of 50 g/L glucose and 0.03 g/L cell biomass at time 0 h in a 1 L batch reactor. The RBatch was specified as a reactive system with corresponding reaction (the FORTRAN user kinetics subroutine name specified under Reactions in Aspen Plus) selected. The batch reactor was run at a constant temperature of 39 ^{o}C and 1 atm pressure, ending at a total fermentation time (reaction time) of 32 h. The initial step size and maximum step size of the integration variable were both set to 0.01 h from their default values of 0.1 h in the RBatch block. In this way, the amount of material transferred to the reactor at the beginning of fermentation from the feed holding tank, the batch charge (calculated as the product of the feed flow rates and the cycle time) corresponds to the initial batch charge of 50 g/L glucose and 0.03 g/L cell biomass.

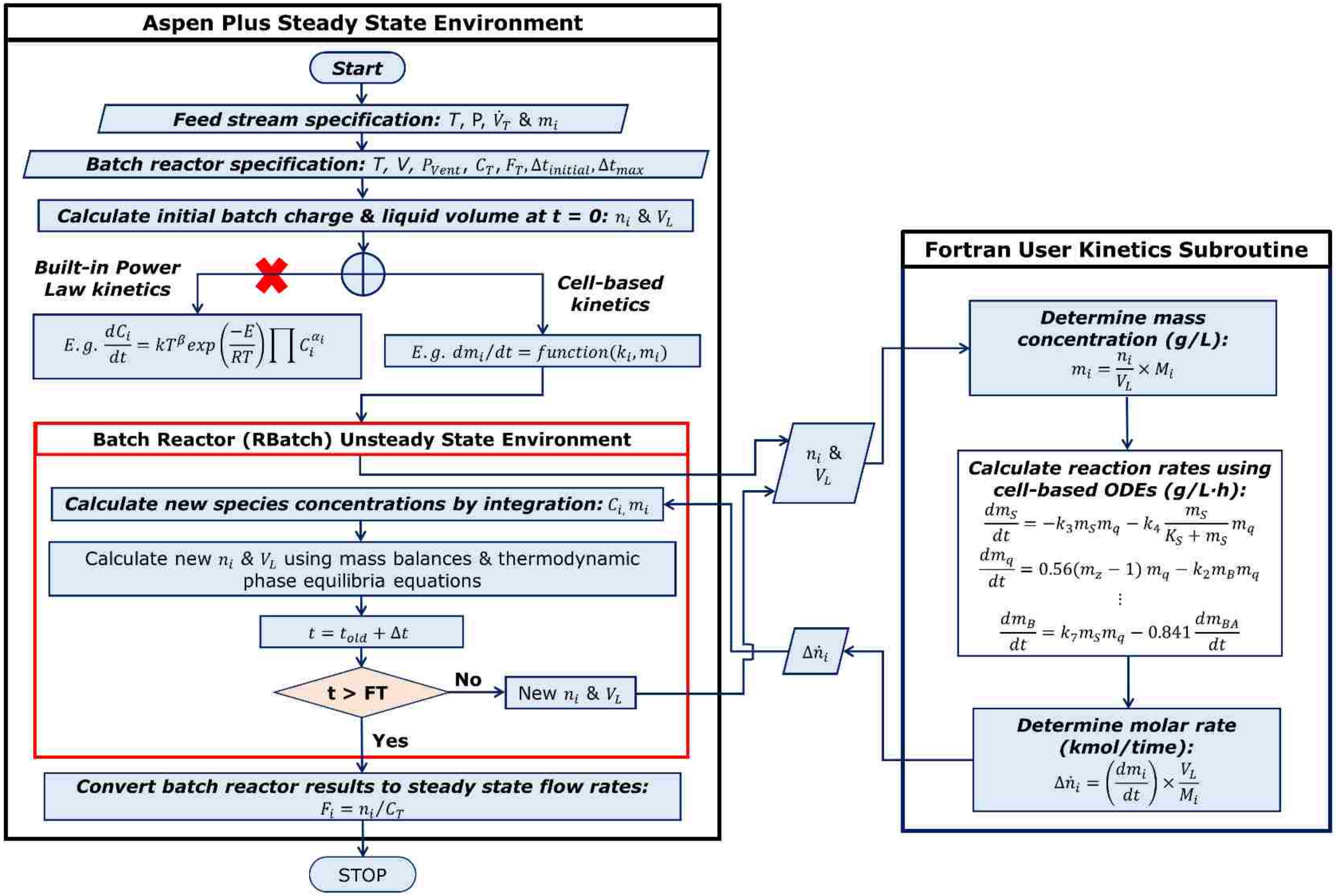

Aspen Plus Steady State Environment
Start
Feed stream specification: T, P, $\dot{V}_T$ & m_i
Batch reactor specification: T, V, P_{Vent}, C_T, F_T, $\Delta t_{initial}$, Δt_{max}
Calculate initial batch charge & liquid volume at t = 0: n_i & V_L
Built-in Power Law kinetics
$E.g.\ \frac{dC_i}{dt} = kT^{\beta} exp\left(\frac{-E}{RT}\right)\prod C_i^{\alpha_i}$
Cell-based kinetics
$E.g.\ dm_i/dt = function(k_i, m_i)$
Batch Reactor (RBatch) Unsteady State Environment
Calculate new species concentrations by integration: C_i, m_i
Calculate new n_i & V_L using mass balances & thermodynamic phase equilibria equations
$t = t_{old} + \Delta t$
t > FT
No
New n_i & V_L
Yes
Convert batch reactor results to steady state flow rates: $F_i = n_i/C_T$
STOP
n_i & V_L
$\Delta\dot{n}_i$
Fortran User Kinetics Subroutine
Determine mass concentration (g/L): $m_i = \frac{n_i}{V_L} \times M_i$
Calculate reaction rates using cell-based ODEs (g/L·h): $\frac{dm_S}{dt} = -k_3 m_S m_q - k_4 \frac{m_S}{K_S + m_S} m_q$
$\frac{dm_q}{dt} = 0.56(m_z - 1)\, m_q - k_2 m_B m_q$
$\vdots$
$\frac{dm_B}{dt} = k_7 m_S m_q - 0.841 \frac{dm_{BA}}{dt}$
Determine molar rate (kmol/time): $\Delta\dot{n}_i = \left(\frac{dm_i}{dt}\right) \times \frac{V_L}{M_i}$

Figure 4.2 Flow chart showing the interface between the Aspen Plus steady state environment and interaction between the unsteady state batch reactor linked to Fortran user subroutine

The notation is: components (i), temperature (T), pressure (P), $\dot{V}_T$ (total volumetric flow rate of feed stream, L/time), m (mass concentration, g/L), P_{vent} (pressure at which venting begins), C_T (total cycle time), F_T (total fermentation time), $\Delta t_{initial}$ (initial time step size), Δt_{max} (maximum time step size), n (moles), V_L (liquid volume in reactor, L), C (molar concentration, mol/L), E (activation energy), R (universal gas constant), α (order of reaction), β (temperature exponent), M (molar mass, g/mol), $\Delta\dot{n}$ (change in the molar rate, mol/time), k, k_i, k_2, k_3, k_4, k_7, K_S (kinetic parameters), S, X, B, BA (glucose, cells, butanol, butyric acid), Z (physiological marker), t (current time), Δt (variable time step), F (molar flow rate, mol/time).

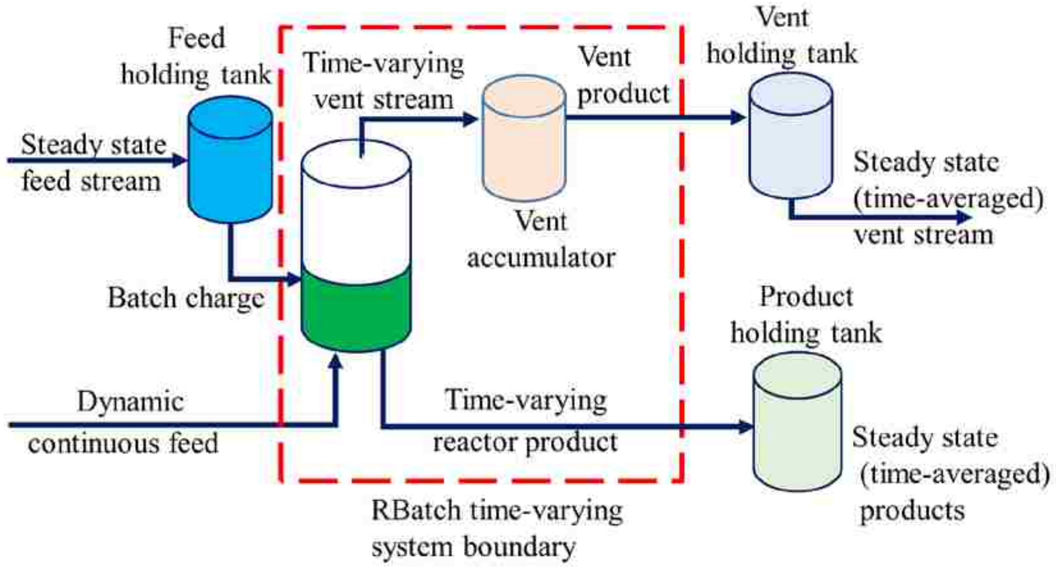


Figure 4.3 The configuration of the RBatch block in Aspen Plus. Adapted and modified from Aspen Plus V8.8 Help.[122]

4.3.2.2 Incorporating biological ODEs into the kinetics of the batch reactor in Aspen Plus

The RBatch unit operation in Aspen Plus can only handle rate-based reactions. For rate-based reaction kinetics, the reaction rate can be calculated using the built-in power law model (eq 4.1) which is an algebraic expression that relates the reaction rate to the reaction rate constants and a product of the concentration of components. The system of ODEs describing the ABE batch fermentation expresses the reaction rates in terms of constant reaction rate parameters and a complex dependence on concentration of components, (for example, eq 4.2 describing glucose consumption rate) and does not conform to the standard built-in power law model. Therefore, a user kinetics subroutine written in FORTRAN (calculating the rates of production/consumption of each component), was dynamically linked to the batch reactor (RBatch) in Aspen Plus to simulate the unsteady state batch ABE fermentation process. The FORTRAN user kinetics subroutine was written based on the ODEs of the selected fermentation model (described in section 4.3.1.1), compiled into a written subroutine (creating a readable Aspen Plus file from the written subroutine) and supplied as a compiled readable file to Aspen Plus to run the simulation dynamically (Appendix C and D).

Power Law Model: $r = kT^n \exp(-E/RT) \prod (C_i)^{\alpha_i}$ (4.1)

ODE describing the consumption of glucose: $\frac{dC_S}{dt} = -k_3 C_S C_X - k_4 \frac{C_S}{K_S + C_S} C_X$ (4.2)

where $r, k, T, n, E, R, \alpha, i, \prod, C, Cs, C_X$, and ($k_3, k_4, K_S$) are rate of reaction, pre-exponential factor, temperature, temperature exponent, activation energy, universal gas law constant, concentration exponent, component index, product operator, component concentration, substrate concentration, cell biomass concentration and kinetic parameters, respectively.

4.3.2.3 Communication between the batch reactor and user kinetics subroutine and running the RBatch in Aspen Plus

The rates of production/consumption of each component in the batch reactor are calculated in a user kinetic subroutine written in FORTRAN linked to the RBatch. In order to calculate the rates at the current fermentation time (*t*), the ODEs require the rate kinetic parameters and the concentration of each component. The moles of each component in the fermentation mixture and volume of liquid components in the reactor are passed from the RBatch to the subroutine at each time-step. In the subroutine, the concentration of each component in g/L (calculated from the moles of each component and the liquid volume) is substituted into the ODEs describing the ABE fermentation to calculate the rate of consumption/production of each component (g/L.h). The reaction rates are reconverted to a form that can be passed to the RBatch reactor (from g/L.h to kmol/s). RBatch then uses the variable-step-size Gear algorithm as the integration method to solve for the new concentrations of components at the current fermentation time (*t*). The phase composition, concentrations of each component in the reactor (vapor, liquid, solid) and the stripped stream (vapor), and other estimated properties including the volume of the liquid, solid and vapor components are evaluated using the thermodynamic model (section 4.3.1.3) to satisfy material and energy balances.

If the current integration time is less than the total fermentation time specified for the batch reactor, the new calculated values (moles of each component and the liquid volume) are passed from the RBatch to the subroutine for the next step calculation until the end of the total fermentation time specified. The RBatch unit operation is able to generate time-dependent data for a batch fermentation process in Aspen Plus because of the integration process. The total accumulated material in the reactor and the vent accumulator at the end of fermentation are converted into steady state flow rates, calculated as the ratio of the total accumulated mass in the vent accumulator or the reactor at the end of fermentation to the total cycle time. The vent product stream is the contents of the vent accumulator at the end fermentation. The contents of the vent accumulator is a continous time-varying vapor that leaves the reator.[122]

4.3.2.4 MATLAB simulations to verify unsteady state Aspen Plus ABE batch fermentation results

To verify the Aspen Plus procedure developed, the simulation results of the ABE fermentation process using the RBatch block in Aspen Plus (time-dependent concentrations of the substrate, intermediates and products) were compared to integration results obtained from MATLAB. The batch reactor (RBatch block) in Aspen Plus solves the mass, energy and composition equations for each fermentation time step using the variable-step-size Gear algorithm as the integration method.[122] Ode15s in MATLAB is a variable-step and variable-order solver that can be set to use the backward differentiation formulas (BDF), also known as the Gear's method. MATLAB ode15s was, therefore, set to use the BDF (Gear's method) with the corresponding integration parameters used in the RBatch block so that the same integration method was used in both MATLAB and Aspen Plus. The liquid phase fermentation broth in the Aspen Plus batch reactor (in the absence of a gas stripping process) is minimally affected by thermodynamic contributions, such that the MATLAB results can be used to verify the procedure developed for RBatch in Aspen Plus without gas stripping.

4.3.3 Unsteady state fermentation and *in situ* gas stripping simulations

To link the unsteady state batch ABE fermentation and *in situ* gas stripping process in Aspen Plus, a feed of nitrogen gas (0.8, 1.6, 3, 5, 6.4 L/min per L of fermentation broth) was fed continuously to the reactor (conditions specified in Section 4.3.2.1 at specified start times (relative to the beginning of the batch fermentation at t = 0 h) with a vent. For the RBatch with a vent, a reactor volume of 1.009 L (allowing for a headspace requirement for vapors) was specified. A vent opening pressure of 1 atm was specified, allowing the RBatch block to calculate the reactor pressure. Similar to integrated batch ABE fermentation and *in situ* gas stripping laboratory experiments, where gas stripping is initiated after a specified batch fermentation time (*e.g.*, 20 h[1]) or product concentration in the fermentor (*e.g.*, 3 – 4 g/L of ABE[42]), gas stripping was started after 15 h of fermentation when the ABE concentration was about 5.7 g/L. For simplicity, it was assumed that there was a complete recovery of the stripped liquid components (acetone, butanol, ethanol, acetic acid, butyric acid, and water in the condensate) in the simulation of the unsteady state batch fermentation and *in situ* gas stripping.

4.3.4 Simulation of separate steady state fermentation and gas stripping processes

The state-of-the art simulation of the batch fermentation and *in situ* gas stripping in Aspen Plus is currently the separate steady state batch fermentation and gas stripping, in which the batch fermentation is simulated with a stoichiometric reactor and the final fermentation broth from the stoichiometric reactor is fed to a flash unit to simulate the gas stripping process. A steady state batch Aspen Plus ABE fermentation with initial conditions of 50 g/L glucose and 0.03 g/L cell biomass was simulated using a steady state stoichiometric reactor (RStoic block). Appendix E presents the stoichiometric equations used in the steady state simulation. The stoichiometric parameters for the simulation were obtained using the final product yields from the RBatch block linked with the FORTRAN user kinetics subroutine (0.319, 0.495, 0.080, 0.120, 0 (mole of product/mole of glucose fed) for acetone, butanol, ethanol, acetic and butyric acids, respectively). The resulting liquid fermentation broth calculated using the steady state RStoic block was fed to an isothermal flash unit (39 °C) with different N_2 gas flow rates (L/min per L of broth) to simulate the gas stripping process. The trends in the results of the steady state separate fermentation and gas stripping simulations were compared with unsteady state ABE batch fermentation and *in situ* gas stripping simulated with the RBatch block linked with the Fortran user kinetics subroutine with initial concentrations of 50 g/L glucose and 0.03 g/L cell biomass with gas stripping (N_2 flow rates of 0.8, 1.6, 3, 5, 6.4 L/min per L of broth) started after 15 h of batch fermentation.

4.3.5 Calculation of ABE fermentation performance parameters

The productivity, yield, percent mass recovery, and selectivity are used to describe the performance and operation of the batch ABE fermentation and the integrated batch ABE fermentation and *in situ* product recovery by gas stripping. The following parameters were calculated, applied to ABE as mixture or individual components:

$$Productivity\ (g/L\ /h) = \frac{MCR+MCV}{VLIQS*t} \tag{4.3}$$

$$Yield\ (g/g) = \frac{MCR+MCV}{GS} \tag{4.4}$$

$$Percent\ recovery\ (\%) = \frac{MCV}{MCR+MCV} * 100 \tag{4.5}$$

$$Selectivity\ = \frac{y(1-x)}{x(1-y)} \tag{4.6}$$

where *MCR* is the accumulated mass in the reactor (grams), *MCV* is the accumulated mass in the stripped stream in grams (vent accumulator, condensate), *VLIQS* is the total volume (L) of the liquid and solids contents in the reactor, *t* is the fermentation time (h). *GS* is the total grams of sugar utilized (calculated as the difference between the initial mass of glucose and the mass of glucose at the end of fermentation), *y* and *x* are the mass fractions in the stripped vapor stream (assuming complete recovery of acetone, butanol, ethanol, water, butyric and acetic acids and neglecting CO_2, H_2 or N_2) and the accumulated mass fraction in the reactor (acetone, butanol, ethanol, water, butyric and acetic acids) at the same time, respectively.

4.4 Results and discussion

4.4.1 Validation of the Aspen Plus unsteady state batch fermentation with MATLAB simulation results

Figure 4.4 illustrates a comparison of the simulation results of the batch ABE fermentation using a FORTRAN user kinetic subroutine linked to the RBatch block in Aspen Plus and the integration of the system of ODEs describing the batch ABE fermentation in MATLAB. The results for all components are provided in Appendix Figure F1. MATLAB and Aspen Plus simulation of batch fermentation in the absence of gas stripping are indistinguishable, with the exception of acetone production. Figure 4.4A depicts a typical Monod-type microbial cell growth kinetics with exponential growth and a stationary phase followed by a death phase, with corresponding consumption of the substrate (Figure 4.4B). Consistent with the metabolic pathway of *C. acetobutylicum*, the concentration profiles for acetic acid (Appendix F, Figure F1F) and butyric acid (Appendix F, Figure F1G) depict a trend of acid production (from 0 to 13 h) followed by consumption and reutilization (13 to 32 h) of these acids to produce solvents. Solvent production (acetone, butanol and ethanol) was therefore not significant until after about 13 h, in support of starting gas stripping after 15 h of batch fermentation. In MATLAB, negative concentrations were predicted for acetone between 0 and 13 h, which are physically unrealistic but present in the ODEs of the fermentation model (Figure 4.4C).

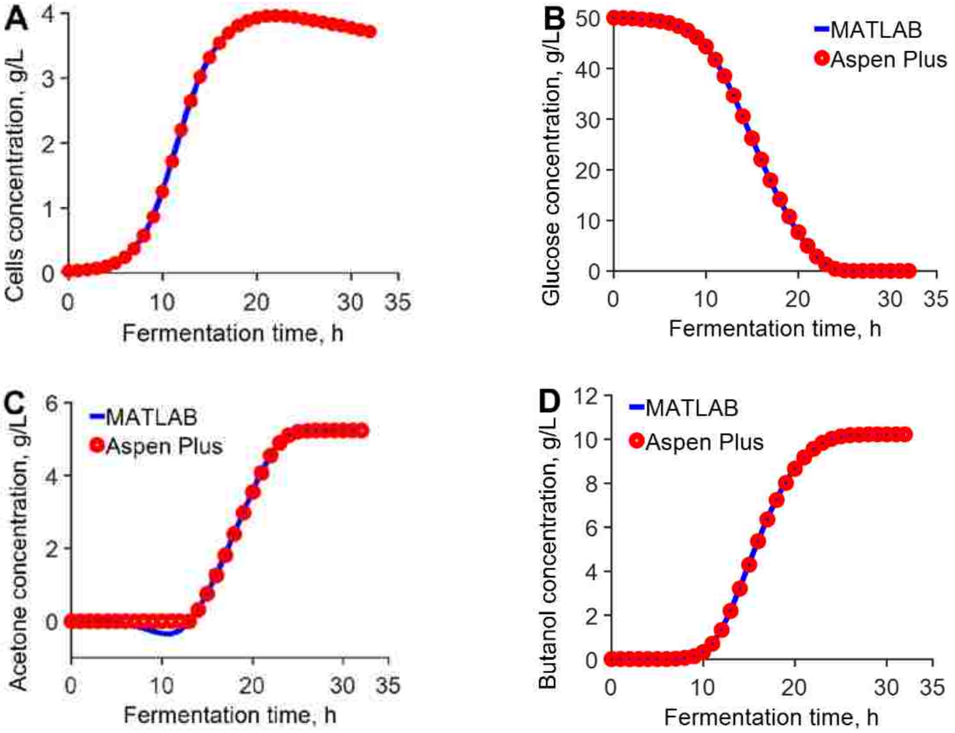


Figure 4.4 Comparison of batch fermentation simulation results in Aspen Plus (RBatch) with the integration of the ordinary differential equations describing the batch fermentation process in MATLAB for cells (A), glucose (B), acetone (C) and butanol (D)

In Aspen Plus, the RBatch block solves the mass and energy component equations to satisfy the material and energy balances, and negative concentrations are avoided. Comparison of the MATLAB and Aspen Plus results validates the direct use of the ODEs incorporating the autocatalytic production of cells, substrate consumption and production and inhibition of fermentation products in Aspen Plus to provide time-dependent simulations of batch fermentors.

4.4.2 Effect of gas flow rate on the fermentor and condensate ABE concentrations, total ABE produced, productivity and yield

Figure 4.5 shows the concentrations of acetone, butanol and ethanol in the fermentor and condensate (stripped stream) in the batch ABE fermentation with gas flow rates of 1 and 5 L/min per L of broth for the simulation of batch fermentation with *in situ* gas stripping. The gas-liquid partitioning behavior of the acetone/ethanol/butanol/water mixture is complex.[7] In the simulated process, an increase in gas flow rate from 1 to 5 L/min per L of fermentation broth resulted in a decrease in fermentor concentrations of ABE compared with the respective fermentor concentrations without gas stripping, indicating the successful removal of the volatile ABE fermentation products. However, stripping of water, the most abundant volatile component in the fermentation broth, contributes to the decrease in the ABE concentration in the condensate with increasing gas flow rates. Thus, there exists a tradeoff between product recovery from the fermentation broth and ABE concentration in the corresponding condensate when selecting a gas flow rate. The concentrations of acetone, butanol and ethanol in the fermentor are lower in the integrated batch fermentation and *in situ* gas stripping compared to the batch fermentation simulation alone (Appendix F, Figure F1).

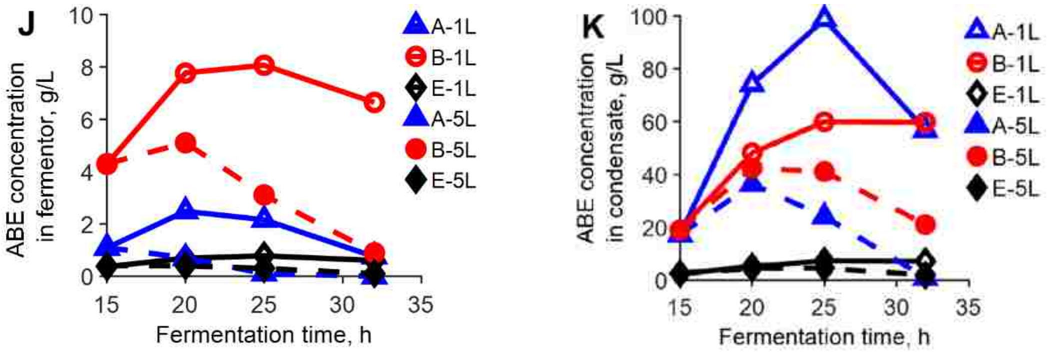


Figure 4.5 Comparison of the concentrations of acetone (A), butanol (B) and ethanol (E) in the fermentor (J) and condensate (K) using different gas flow rates
The reactor volume was 1 L and gas stripping started after 15 h. The notation is 1L (corresponding to 1L/min N_2 per L of broth) and 5L (corresponding to 5L/min N_2 per L of broth).

A similar trend was observed by Ezeji et al [42] using an initial glucose concentration of 60 g/L with a gas recycle rate of 3 L/min per L of fermentation broth (started after 15 h of fermentation). In their batch fermentation study, the acetone, butanol and ethanol concentrations in the fermentor decreased from 5, 12, 1 g/L without gas stripping to 4, 6 and 0.1 g/L with gas stripping, respectively. Similarly, Xue et al.[123] observed an increase in the concentrations of acetone, butanol and ethanol in the stripped stream compared with their respective concentrations in the fermentor using an initial glucose concentration of 80 g/L and a gas flow rate of 1.5 L/min per L of the fermentation broth.

The *in situ* ABE recovery by gas stripping coupled with the batch fermentation has been shown to reduce butanol toxicity, and improve the performance of the ABE fermentation process.[42] The total ABE produced (total concentration), productivity and yield for an integrated ABE batch fermentation and *in situ* gas stripping with 0 (no gas stripping), 0.8, 1.6, 3, 5 and 6.4 L/min N_2 per L of fermentation broth were simulated using the Fortran user kinetics subroutine linked with the RBatch block in Aspen Plus (Figure 4.6). Compared with the simulated results of the batch ABE fermentation without gas stripping, the total concentration of the ABE produced, productivity and yield were improved up to 105, 110, 119, 130 and 150% for the integrated batch process employing 0.8, 1.6, 3, 5 and 6.4 L/min N_2 per L of broth, respectively. The improvement in the total ABE produced and productivity in this study are comparable to the 133 and 210% enhancements in the total ABE produced and productivity, respectively, reported by Ezeji et al.[42] in their laboratory integrated batch ABE fermentation and *in situ* gas stripping with an initial glucose concentration of 60 g/L and a gas flow rate of 3 L/min per L of broth started after 15 h. A total ABE concentration of 17.7 g/L and 23.6 were reported by Ezeji et al.[42] in their laboratory batch ABE fermentation and without and with *in situ* gas stripping, respectively. Similarly, the simulation of an integrated ABE batch fermentation and *in situ* gas stripping with 0 (no gas stripping), 0.8, 1.6, 3, 5 and 6.4 L/min N_2 per L of fermentation broth predicts a total ABE produced of 16.5, 17.3, 18.1, 19.6, 21.4 and 24.6 g/L, respectively.

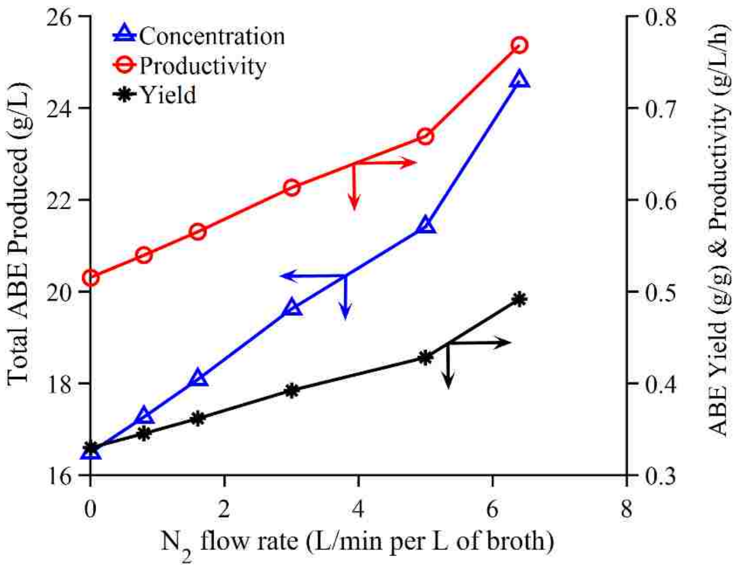


Figure 4.6 Total ABE produced (total concentration), productivity and yield from the simulated batch ABE fermentation and *in situ* gas stripping with different gas flow rates.

4.4.3 Comparison of performance of batch ABE fermentation and *in situ* gas stripping simulations to available literature.

A broad range of gas flow rates relative to fermentation broth volume, stripping or operating temperatures, and initiation times of gas stripping have been investigated experimentally for the gas stripping of ABE fermentation.[38] The models used to describe solventogenic *Clostridia* species (for example, *C. acetobutylicum*, *C. beijerinckii*, *C. saccharobutylicum* and *saccharoperbutylacetonicum*[4-5]) generally lack applicability to other microorganisms, making direct comparisons of *in silico* analyses and available laboratory ABE fermentation experimental data difficult. To systematically analyze the effect of the broad range of gas flow rates employed in the ABE batch fermentation and *in situ* gas stripping, the trends in the simulation results were therefore compared with the observed trends in literature. The batch ABE fermentation is characterized by low product concentration (< 20 g/L ABE), low reactor productivities (< 0.3 g/L/h) and low ABE yield (0.28 – 0.33 g/g) as a result of product toxicity (especially due to butanol concentrations > 13 g/L) to the microorganisms used in fermentation.[11] Figure 4.7 shows the ABE productivity, yield, total ABE produced, selectivity versus the gas flow rate per L of broth from available literature data and data predicted from the Aspen Plus RBatch unsteady state simulation for batch fermentation and *in situ* gas stripping (gas flow rates of 0.8, 1.6, 3, 5, 6.4 L/min per L of broth).

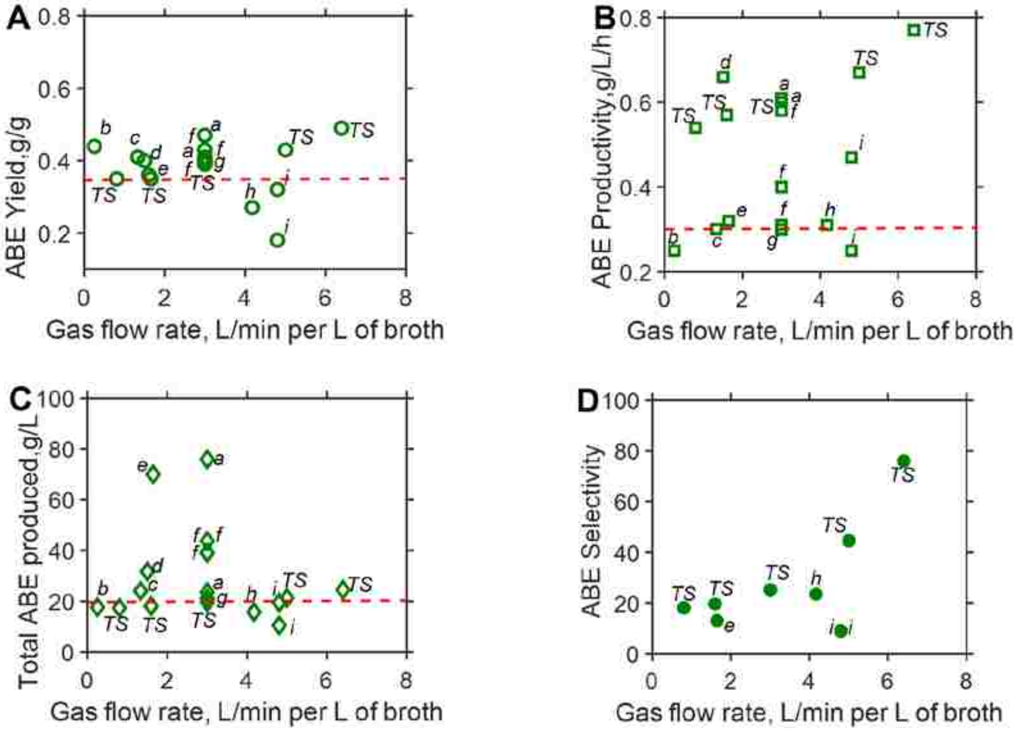


Figure 4.7 ABE yield (A), ABE productivity (B), total ABE produced (C) and ABE Selectivity (D) versus normalized gas flow rates (L/min per L of fermentation broth) from batch fermentation and *in situ* gas stripping literature data from refs *b*,[124] *c*,[125] *d*,[123] *e*,[126] *a*,[42] *f*,[127] *g*,[128] *h*,[129] *i*,[130]Aspen Plus unsteady state batch fermentation and *in situ* gas stripping simulation (*TS*) and performance benchmark (- - - - - -) for a typical batch fermentation of 0.35 g/g ABE yield, 0.30 g/L/h ABE productivity and 20 g/L total ABE produced (chosen based on data from Qureshi and Blaschek)[11]

Generally, about 3 L/min per L of fermentation broth is the gas flow rate used most in batch fermentation and *in situ* gas stripping experiments while the lowest and highest gas recycle rates used are 0.25 and 4.8 L/min per L of broth. At low gas flow rates, the ABE yield (Figure 4.7A), ABE productivity (Figure 4.7B), total ABE produced (Figure 4.7C) and ABE selectivity (Figure 4.7D) for the literature data increases (significantly above their respective limits in batch fermentations) with gas flow rate up to about 3L/min per L of broth when these parameters are observed to have optimum values (a range of values observed at 3 L/min per L of broth). At higher gas flow rates, the ABE performance decreases with increasing gas flow rate. The performance of the integrated ABE fermentation and *in situ* gas stripping is consistent with literature data up to about 3 L/min per L of broth. Above 3L/min per L of broth, the ABE productivity, ABE yield, total ABE produced predicted from the Aspen Plus simulation is significantly higher than the batch ABE fermentation benchmark productivity of 0.30 g/L/h, above 0.35 g/g ABE yield and approximately equal to 20 g/L total ABE produced, respectively.

4.4.4 Comparison of traditional Aspen Plus batch separate steady state fermentation and gas stripping with unsteady state batch fermentation and *in situ* gas stripping simulations.

Previous Aspen Plus simulations of the batch ABE fermentation and *in situ* gas stripping have been decoupled and described using steady state equilibrium-based gas stripping of ABE fermentation broths representing the final fermentation concentrations. To compare this approach with our unsteady state ABE fermentations with *in situ* gas stripping, we simulated a steady state ABE fermentation using a stoichiometric reactor with stoichiometric coefficients of 0.319, 0.495, 0.080, 0.120, 0 (mole of product/mole of glucose fed) for acetone, butanol, ethanol, acetic and butyric acids, respectively, calculated from the model of Votruba et al.[120] Steady state gas stripping of the fermentation broth, which had a final composition of 5.1, 10.2, 1.0, 2.0, 0 and 0 g/L for acetone, butanol, ethanol, acetic acid, butyric acid and glucose, respectively, was simulated with an isothermal flash unit at 39 ^{o}C using different flow rates of N_2 (0.8, 1.6, 3, 5, 6.4 L/min per L of broth). The stoichiometric equations used are presented in Appendix E. The trends from the steady state simulation are compared with the integrated batch ABE fermentation and *in situ* gas stripping for initial concentrations of 50 g/L glucose and 0.03 g/L cell biomass at 39 ^{o}C with gas stripping started after 15 h of fermentation (Figure 4.8).

In the steady state simulation, the selectivities (Figure 4.8M) and condensate concentrations (Figure 4.8O) of acetone, butanol, ethanol and ABE in the condensate are not a strong function of gas flow rate. Condensate refers to the contents of the stripped stream (accumulated in the vent accumulator). The steady state selectivities, percent recovery and condensate concentration of acetone decrease slightly with increasing gas flow rate whereas the selectivities and condensate concentrations of butanol and ethanol increase slightly with increasing gas flow rate (Appendix G, Table G.1). In contrast, the simulation linking batch fermentation with *in situ* gas stripping (unsteady state) predicts that selectivities (Figure 4.8N) and recoveries of acetone, butanol, ethanol and ABE

(Appendix G, Table G.2) increase significantly with increasing gas flow rate per L of the fermentation broth whereas the respective condensate concentrations (Figure 4.8P) decrease with increasing gas flow rate.

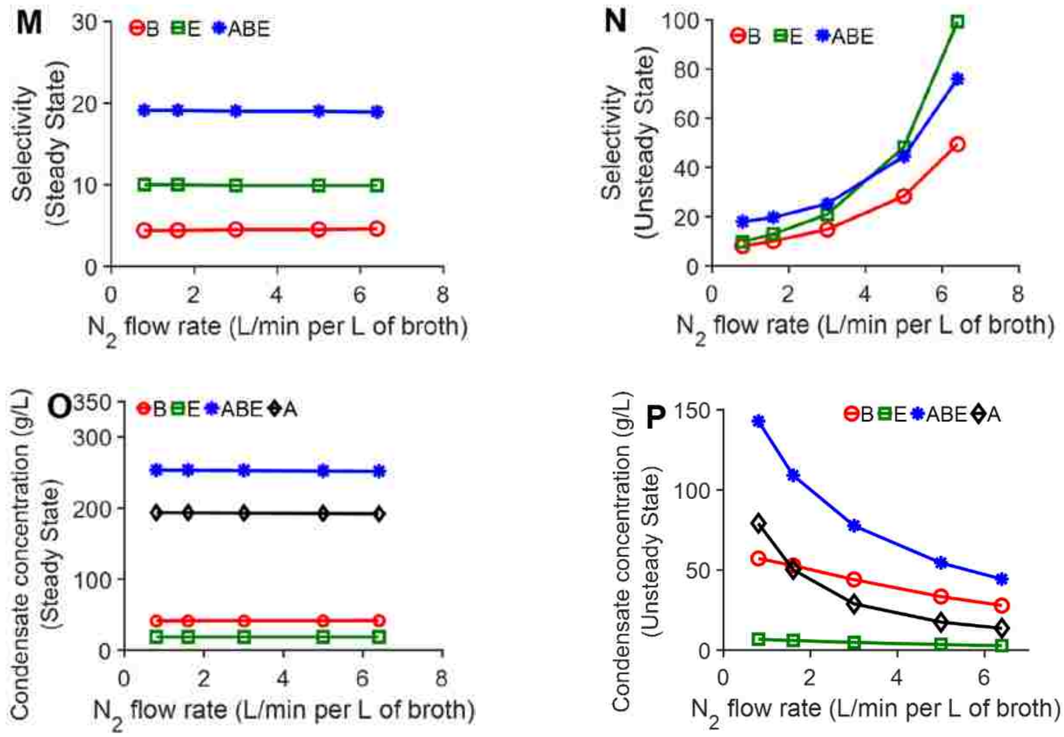


Figure 4.8 Selectivities and condensate concentrations at the end of 32-h batch fermentation and *in situ* gas stripping of acetone (A), butanol (B) and ethanol (E) using traditional steady state (stoichiometric reactor and flash unit) and unsteady state (RBatch) in Aspen Plus with different gas flow rates.

The notations are: selectivity from the steady state simulation (M), selectivity from the unsteady state simulation (N), concentration of the condensate from the steady state simulation (O) and concentration of the products from the unsteady state simulation (P)

The unsteady state selectivity of butanol, ethanol, and the ABE mixture overall increase significantly with increasing gas flow rate whereas the steady state selectivities decrease slightly (with the exception of butanol), but are largely insensitive to gas flowrate. Acetone, which was almost exhausted in the fermentor at high gas flow rates, had significantly higher selectivities (results not shown). In general, high selectivities can be observed at conditions of low product concentration in the fermentor, thus selectivity is sensitive to both the start time of the gas stripping (controlling the initial accumulation of the product) and the gas flow rate in the integrated batch fermentation and *in situ* gas stripping. Correspondingly, the condensate concentrations of butanol, ethanol, acetone, and the ABE mixture decrease significantly in the unsteady state simulation relative to the steady state concentrations, which are relatively insensitive to gas flow rate. Again, this observation can be attributed to the time-dependent decrease in concentration of the ABE fermentation products in the fermentor over the course of gas stripping. In the unsteady state simulation, the product in the condensate is diluted by the significant amount of water that is also volatilized in the gas stripping process, which increases with gas flowrate.

The results of batch fermentation coupled with *in situ* gas stripping simulations show that the gas stripping gas flow rate has a significant effect on the performance of the batch fermentation and *in situ* gas stripping process whereas such trends are not evident in the absence of a time-dependent fermentation model linked directly to the gas stripping process. Simulations and representations of the fermentation coupled with the gas stripping process based on unsteady state models, such as the cell-based dynamic mathematical models, offer opportunities to further investigate and understand the interaction and relationship among the typical parameters (e.g. selectivities, recoveries and condensate concentrations, total ABE produced, productivity and yield) that describe integrated fermentation and *in situ* gas stripping process.

4.5 Conclusion

This study has focused on simulation of batch fermentation as an unsteady state process by incorporating autocatalytic production of cells, time-dependent concentrations of the fermentation components, and substrate and product inhibitions in the framework of Aspen Plus, a universally accepted traditional process simulator of choice for refinery and chemical processes. This simulation approach allowed the batch fermentation process (described using a time-dependent fermentation model) to be coupled with *in situ* product recovery by gas stripping for the first time. In this way, the time-dependent phase composition and concentrations of components in a fermentor (solid, liquid, and vapor) and stripped stream (vapor) were predicted by the thermodynamic models in Aspen Plus to provide realistic simulations of integrated batch fermentation and *in situ* gas stripping experiments under different operating conditions. The performance of the integrated batch and *in situ* gas stripping is shown to be dependent on the gas flow rate employed, an artifact that is absent without a time-dependent fermentation model linked *in situ* to the gas stripping process. While the traditional steady state separate fermentation and gas stripping are useful in studies involving the overall bioprocess, they may be inadequate for systematic analyses of bioprocesses, especially if fermentations are linked with *in situ* separations. Additionally, our simulation approach predicts trends that are consistent with available literature data and offer insight into the performance of the ABE batch fermentation and *in situ* gas stripping at high gas recycle flow rates outside the range investigated in available literature. The simulation approach in this research will allow the full suite of PSE tools to be applied to the ABE production process, providing a decision-support tool to aid the fermentation experimentalist. This research also provides a general platform to integrate biorefinery processes (fermentations) and chemical and refinery processes in the process simulation packages.

Chapter 5

Case study: Application of PSE Tools to the Design of a Fermentation Reactor with *in situ* Gas Stripping

5.1 Introduction

Biofuels, bioethanol, and biobutanol, produced from lignocellulosic biomass as sustainable energy alternatives to fossil fuels[1] have properties that make them excellent liquid transportation fuels.[37-38] Bioethanol and biobutanol can be produced via the ABE fermentation process using microorganisms such as *Clostridium acetobutylicum, C. beijerinckii, C. saccharobutylicum, C. Saccharoperbutylacetonicum.*[4-5] The batch ABE fermentation produces low final ABE titers, yields and productivity[38, 131-133] because of product (mainly butanol) and substrate inhibitions. *In situ* product recovery techniques such as gas stripping, adsorption, etc. are used to alleviate product inhibition.[38, 42] Gas stripping employs an inert gas, such as nitrogen, to remove the ABE solvents selectively as they are produced, preventing the buildup of products to inhibitory levels.[10]

Optimization of integrated batch fermentation and *in situ* gas stripping involves the complex interaction of different process variables. The resultant nonlinear model often involve competing variables. To observe the interaction and trade-off that exist among such competing variables, multiobjective optimization is used.[84] To this end, this case study demonstrates how the novel framework proposed in Chapter 3 is used to optimize the operating conditions (gas flow rate relative to the fermentation volume; and gas stripping initiation times) to maximize the total ABE produced, ABE selectivity and concentration of ABE in the stripped stream (condensate) of an integrated batch and *in situ* gas stripping process. Aspen Plus and MATLAB are used as a model process simulator and rigorous optimization platform, respectively. The multiobjective genetic algorithm in MATLAB is linked through a Microsoft® Component Object Model (COM) interface with the batch reactor in Aspen Plus that has been coupled with a FORTRAN user kinetics subroutine (calculate data required to evaluate the objective function values). The results of the MOO are compared to SOO (weighted average of the objective function values). Figure 5.1 summarizes how the proposed framework is used to link the genetic algorithm optimization toolbox in MATLAB to Aspen Plus.

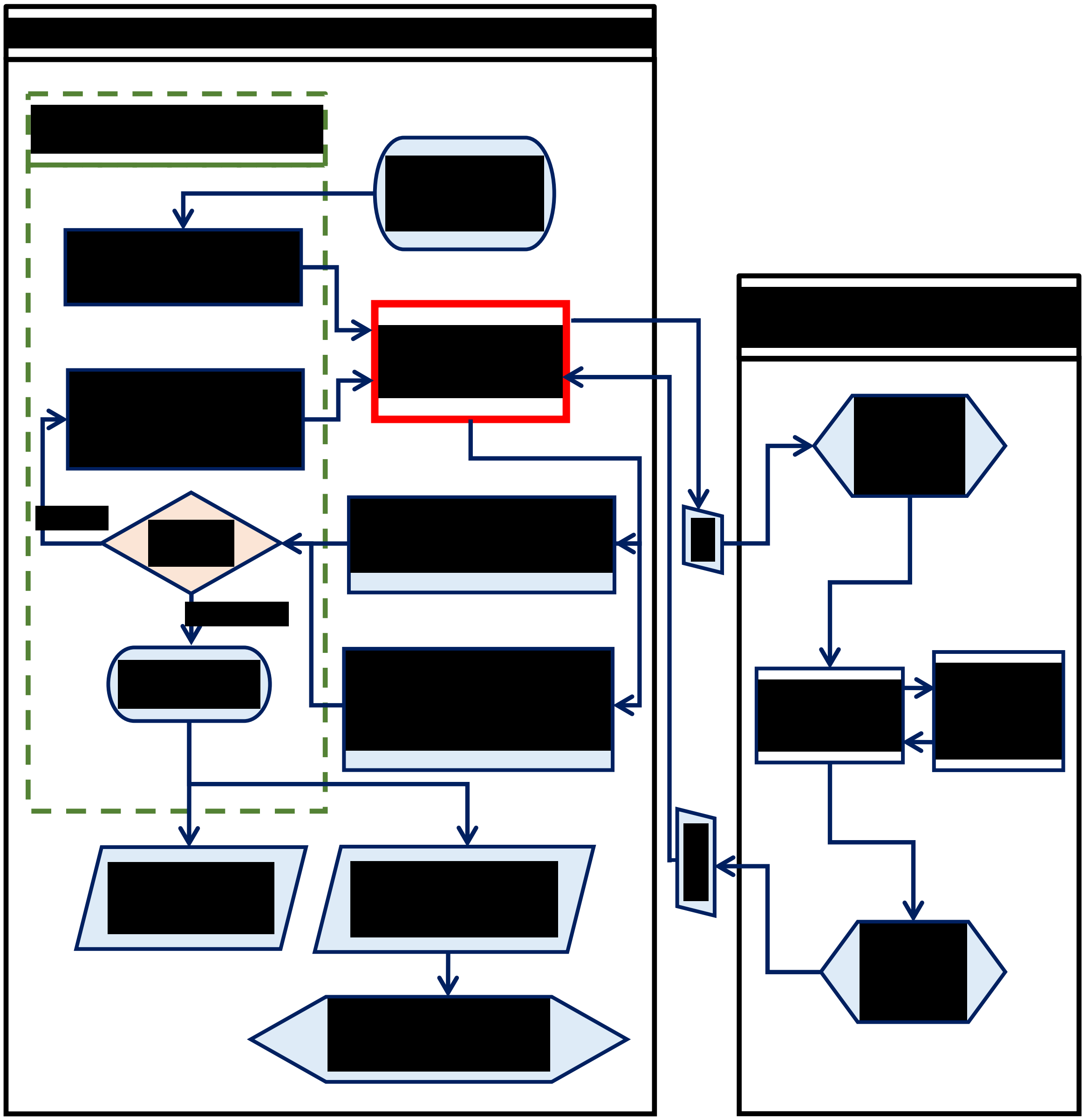


Figure 5.1 Application of the proposed framework to process simulation and optimization of an integrated batch and *in situ* gas stripping

The notations are: decision variables (*DV*), number of decision variables (*NV*), lower bounds of decision variables (L_b), upper bounds of decision variables (U_b), population size (*PS*), stopping criteria (*SC*), objective functions (*O*), component object module (COM), objective function values (*Z*), relative weight of each objective function (*w*), number of non-dominated solutions (*j*) and population (*P*).

5.2 Process optimization

The genetic algorithm (ga) optimization in MATLAB minimizes an objective (using *ga* solver) or sets of objective functions (using *gamultiobj* solver) subject to a set of constraints (decision variables). For the integrated batch and *in situ* gas stripping process in this case study, the objective was to maximize the ABE concentration in the stripped stream, ABE selectivity and total ABE produced subject to the gas stripping initiation times after batch fermentation started and the gas flow rate employed per liter of the fermentation broth. To initiate the algorithm, the decision variables (gas flow rate and gas stripping start times), number of decision variables (2), lower (0.1 L/min per L broth and 15 h) and upper (5 L/min per L broth and 25 h) bounds of decision variables, population size (500), stopping criteria (change in the spread of the Pareto front less than the typical uncertainty in the measurement of the ABE parameters in the laboratory experiments)[1] and the objective functions expressions are specified.

MATLAB supports the creation of a Microsoft® Component Object Model (COM) automation server for a "controller" external program through the actxserver.[134] An external program such as Aspen Plus can be controlled (opened, data written to and read, saved and closed) through the COM interface created. Through the MATLAB COM interface, each element in the population is passed from MATLAB to Aspen Plus (run the batch reactor in Aspen Plus) for data to be used to evaluate the objective function values corresponding to each set of decision variables in the population. The objective function values (solution space) are then scored and ranked in the in the genetic algorithm. Based on the scores and rank, the decision variables in the current population with the best scores are selected to be parents of the next generations. The remaining individuals required to form the total number of population (based on the population size) are randomly generated. The new population (new generation) generated is passed to Aspen Plus to determine objective function values of the current population. Optimization is terminated when the stopping criteria is met.

For the multi-objective optimization (eq J.1 in Appendix J), a set of equally optimal non-dominated solutions (set of decision variables, DV_j, and their corresponding objective function values, Z_j, where j is the number of non-dominated solutions) are generated. A non-dominated solution is one in which at least one of the objective function values is better when compared to another solution. There are several methods such as the NFM and RSM to score and rank the solution of equally optimal solutions to find the best point. The NFM, used in this case study, incorporates the expertise of the decision maker in terms of relative weights of each objective function, indifference, preference and veto thresholds.

The details of the NFM are presented in Appendix K. Equal weights were set for all objective functions and 5 %, 10 % and 30 % of the range of the optimal solutions for each objective function used for the indifference, preference and veto thresholds, respectively. For a single-objective optimization approach (eq J.2 in Appendix J), one unique optimal solution (a decision variable, DV and the corresponding objective function values, Z) is obtained. The relative weights (equal weights for all objective functions) used to aggregate the objective functions into a single objective function in the SOO were the same used to relative ranking the Pareto equally optimal solutions from the MOO using the NFM. The general optimization procedure developed was used to optimize the gas flow rate (0.1 – 5 L/min per L of broth, selected based on the normal range of gas flow rates studied in integrated batch ABE fermentation and *in situ* gas stripping)[38] and gas stripping initiation times of 15 to 25 h after, the start of the batch fermentation.[1] See MOO and SOO MATLAB codes in Appendix O.

5.2.1 Fermentation model used in the simulation

The mechanistic model developed by Votruba et al.,[120] a system ODEs describing the batch ABE fermentation based on *C. acetobutylicum*, was used in the simulation. The cell growth model was modified to directly use a Monod-type of substrate dependency and butanol inhibition to realistically utilize substrates and incorporate butanol inhibition (eq 5.1). The only new parameter (*NPAM* in eq 5.1, reevaluated as 0.4892) was determined by minimizing the sum of the squared error between four sets of data for a batch culture of *C. actetobutylicums*[135] and the model predictions from the solution of the ODEs in MATLAB using ode15s. The equation describing the physiological marker in the original model became redundant and not used. All other equations and kinetic parameters in Votruba et al.[120] for glucose, butyric acid, acetic acid, acetone, butanol, ethanol, carbon dioxide and hydrogen were used as presented by the authors.

$$dC_{BIO}/dt = (NPAM\, C_S/(K_S + C_S))\,(K_I\, C_{BIO}/(K_I + C_B)) - k_2 C_B C_{BIO} \tag{5.1}$$

Notation is: concentration in g/L (C), glucose (S), cell biomass (BIO), butanol (B), K_S, K_I, k_2 are all kinetic parameters defined in Votruba et al.[120] and $NPAM = 0.4892\ h^{-1}$.

5.2.2 Batch fermentation simulation in Aspen Plus

The ODEs describing the batch ABE fermentation does not fit the standard power law kinetics traditionally built-in in Aspen Plus. Cells are not found in the databank of

Aspen Plus. To circumvent these challenges, a Fortran user kinetics subroutine (use ODEs to calculate the rates of consumption/generation of each component) linked to the batch reactor and component substitution (simulate cells) were used. Cells were simulated as solids with the physical properties of water from the database of Aspen Plus. The thermodynamic model selected for the simulation is nonrandom two-liquid – Hayden O'Connell (NRTL-HOC) property model with CO_2, H_2 and N_2 declared as Henry components. A typical initial batch charge of 50 g/L glucose and 0.03 g/L cells were used with a fermentation time of 32 h. N_2 gas was fed continuously starting at designated times to simulate gas stripping (remove the volatile ABE components). The ODEs were coded in a Fortran subroutine in Microsoft® Visual Studio 2013, compiled with the Intel® Fortran Parallel Studio XE 2015 Composer Edition for Windows and saved in the same file as the Aspen Plus files using Aspen Plus V9 (Aspen Technology, Inc., MA, USA). The laptop used was a Dell Precision M4800 with a 64-bit operating system, 16.0 GB RAM, Intel® core™ i7-4910MQ CPU @ 2.90 GHz and Windows 7 Professional.

5.3 Results and discussion

Figure 5.2 shows the final Pareto optimal solutions (scored and ranked using the NFM) from the MOO of the gas flow rate (0.1 to 5 L/min per L of fermentation broth) and gas stripping initiation times (15 to 25 h after the start of fermentation) of an integrated ABE batch fermentation and *in situ* gas stripping process to maximize the total ABE produced, ABE selectivity and the concentration of ABE in the stripped stream (condensate). Figure 5.2A shows the final population, which almost covers the entire decision space. From the final population, non-dominated solutions (Pareto optimal points) were chosen as the decision variables. Each of the elements in the decision variable space (Figure 5.2B) can be mapped to a unique element in the objective function value plots in Figures 5.2C, 5.2D and 5.2E. The total ABE produced, ABE selectivity and the concentration of ABE in the condensate were given equal importance (weights) in ranking the Pareto optimal points using the NFM.

The NFM method was able to find a compromised best optimal decision variable of a gas flow rate of 2.9 L/min per L of fermentation broth and gas stripping initiation time of about 20.5 h, corresponding to a total ABE produced, ABE selectivity and ABE concentration in the condensate of 18.4 g/L, 24.8, 103.3 g/L, respectively.

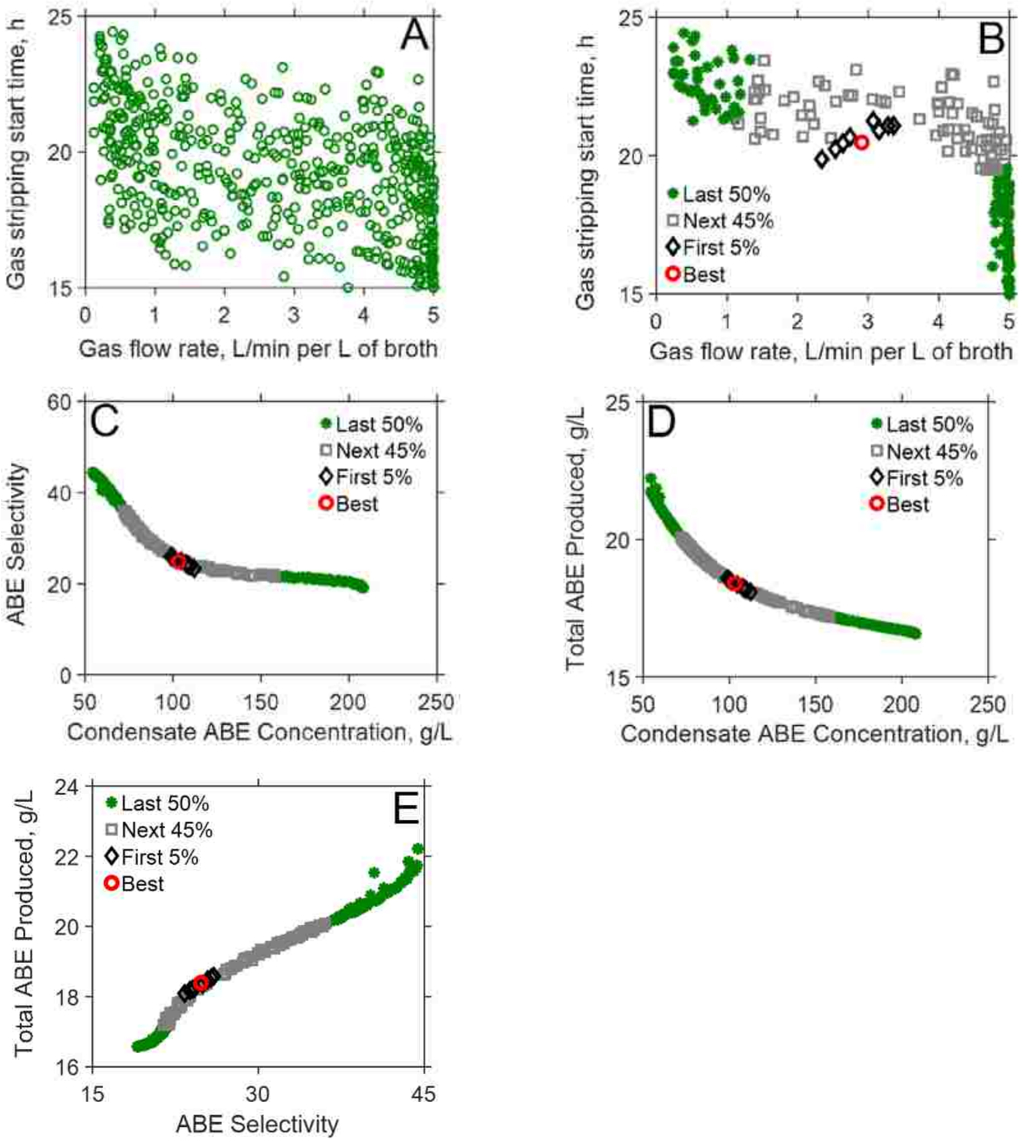


Figure 5.2 Plot of the MOO results for the integrated batch and *in situ* gas stripping process A) Final population B) Decision variables C) ABE selectivity versus ABE concentration in the condensate D) Total ABE produced versus ABE concentration in the condensate E) Total ABE produced versus ABE selectivity. The red circle, black diamonds, grey squares and green asterisk represents the best (top ranked), first 5 %, next 45 % and the last 50 %, respectively, ranked using the NFM method.

The best optimal gas flow rate of 2.9 L/min per L of broth and gas stripping initiation time of 20.5 h agree with the gas flow rate used most in batch fermentation and *in situ* gas stripping experiments of 3 L/min per L of broth (Figure 5.3A versus Figure 5.3 C) and gas stripping initiation times in literature (*e.g.*, 20 h [1]), respectively. In contrast, the SOO approach predicted an extreme very low optimal gas flow rate of 0.3 L/min per L of broth and a higher gas stripping start time of 24 h, resulting in a total ABE produced, ABE selectivity and ABE concentration in the condensate of 16.4 g/L, 19.2 and 208.4 g/L , respectively.

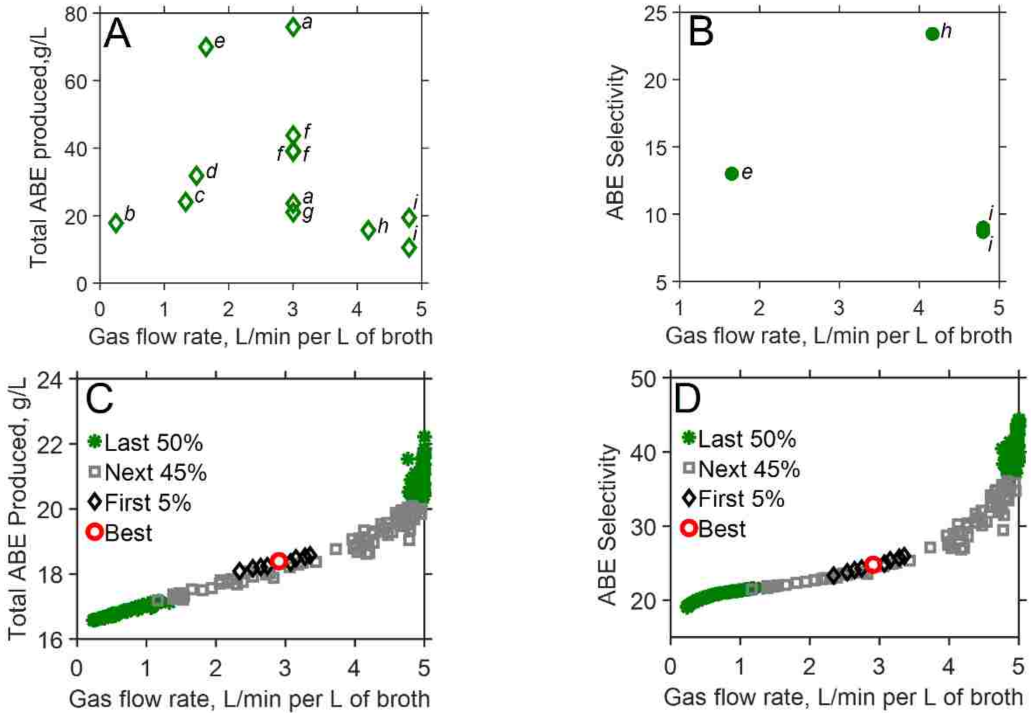


Figure 5.3 Comparison of literature data taken refs *a,*[42] *b,*[124] *c,*[125] *d,*[123] *e,*[126] *f,*[127] *g,*[128] *h,*[129] *i.*[130] with the MOO results from the case study for a batch fermentation and *in situ* gas stripping process.

In general, there is a trade-off when the plot of the total ABE produced versus ABE concentration in the condensate (Figure 5.2D) is compared with the plot of total ABE produced versus ABE selectivity (Figure 5.2E); an increase in total ABE produced leads to a decrease in ABE concentration in the condensate and an increase in the ABE selectivity and vice versa. If the ABE selectivity is improved, the condensate concentration decreases at a lower gas flow rate and late gas stripping initiation times (trade-off). At a higher gas flow rate and early gas stripping initiation time, improving the ABE concentration in the condensate does not have any significant effect on the ABE produced and ABE selectivity. The plots of ABE selectivity versus ABE produced and total ABE produced versus ABE concentration in the condensate are not a competing pair when compared, as an improvement in both ABE selectivity and total ABE produced leads to an improvement in the ABE concentration in the condensate and vice versa.

Literature data reported on ABE selectivity for a batch fermentation and *in situ* gas stripping are limited as seen in Figure 5.3B. The results of this study can therefore shed more light on the effect that gas flow rate and gas stripping initiation times have on the ABE selectivity and the interaction and trade-off that exist among the ABE selectivity and other performance parameters, such as the total ABE produced. With a SOO approach, it would be impossible to observe the interaction and trade-off among the competing parameters and understand how the operating conditions of the process affects ABE performance parameters, especially parameters with very limited reporting such the ABE selectivity.

5.4 Conclusion

In this case study, the commercial PSE tools, Aspen Plus and MATLAB, were integrated using a proposed framework and applied to the simulation of a batch fermentation process to illustrate how the operating conditions of an integrated batch and *in situ* product recovery can be optimized to maximize the performance of the process. These PSE tools allowed the interaction and trade-off that exist among different performance variables (some competing) to be observed. The MOO approach presents many equally optimal options that can be used to understand how deviations in process inputs affect the performance of a process and support the understanding of literature data. With the framework proposed in Chapter 3 and demonstrated herein, the fermentation experimentalist can know the most important experimenter to conduct and save time and resources.

Chapter 6

Multi-objective versus Single-objective Optimization of the Batch Bioethanol Production Based on a Time-dependent Fermentation Model

6.1 Summary

Fermentation microorganisms are susceptible to both substrate and product inhibitions, resulting in dilute aqueous fermentation broths with consequential significant downstream separation costs. This work uses a novel integrated platform of the robust genetic algorithm optimization in MATLAB linked with an Aspen Plus unsteady state batch fermentation simulation to optimize the batch ethanolic fermentation process with respect to initial substrate concentration, fermentation time and *in situ* product removal. A time-dependent fermentation model that utilizes both glucose and xylose, the major sugars present in lignocellulosic hydrolysate, with Monod-type of cell growth, substrate and product inhibitions, is used as a model system. The optimized design variable (initial substrate concentration) from a multi-objective optimization (MOO) and single-objective optimization (SOO) suggests the typical concentrations of sugars from lignocellulosic hydrolysate must be concentrated to optimize the performance of the batch fermentation process. Furthermore, integrating the batch fermentation with an *in situ* product recovery allowed higher initial sugars concentrations to be used in the fermentation process (about 55%, for the best optimal solution in the MOO). This resulted in 16% ethanol productivity (total ethanol produced per batch fermentation time), 143% total ethanol produced and 62% fraction of sugar converted improvements relative to the batch ethanolic fermentation without product recovery. Unlike the single optimal solution obtained in the SOO, MOO presents many equally optimal solutions that can be used to reveal the trade-off and interactions among competing process objectives and as a decision-support tool to guide the choice of design variables and conditions for optimum process performance.

6.2 Introduction

Sustainable resources, such as lignocellulosic biomass, are researched continually for the production of biofuels (biobutanol and bioethanol) as alternatives to fossil fuels.[1, 37-39] The biochemical or sugar platform is a bioprocess in which microorganisms are used to convert a substrate into products[2] by fermentation. Microorganisms used in fermentation on the sugar platform include *Saccharomyces cerevisiae* (yeast, the most commonly used microorganism in batch ethanolic fermentations), *Zymomonas mobilis*, *Escherichia coli, Clostridium thermocellum* as well as filamentous fungi such as *Aspergillus sp*. etc.[3] These microorganisms are usually susceptible to both (high) substrate concentrations and product inhibition. For example, glucose concentrations greater than 150 g/L inhibited ethanol production by *Saccharomyces cerevisiae.*[8] This supports previous reports that substrate inhibition of yeast occurred in the range of 15 – 25% (w/v) sugars concentration, with complete microbial growth inhibition above 40% (w/v) glucose.[8] Also, greater than 5% ethanol concentrations inhibits yeast in ethanol fermentation from glucose.[6] For *Z. mobilis* ZM4, greater than 22 g/L ethanol inhibited cell growth and greater than 86 g/L of ethanol completely ceased cell growth. Similarly, the threshold ethanol concentration above which ethanol inhibition started and the maximum ethanol concentration above which ethanol production ceased were reported as 55 g/L and 127 g/L, respectively for the *Z. mobilis* ZM4.[9] Among the microorganisms for bioethanol production, *Z. mobilis* shows superior properties: it can tolerate ethanol concentrations up to 120 g/L, produce 5 – 10% more ethanol per glucose fermented[3] and has 3 – 5 fold times higher ethanol productivity[136] relative to yeast. Through genetic modifications, *Z. mobilis* is able to utilize glucose and xylose, the two principal sugars in lignocellulosic biomass hydrolysate. *Z. mobilis* uses xylose at a slower rate compared to glucose.[137]

The energy requirements for concentrating and purifying aqueous fermentation broths become significantly higher at alcohol concentrations less than 4% wt.[7] The fermentation step, then, is a crucial step, where fermentation titers, yields and productivities can be improved to reduce the downstream separation cost. The specific challenges in the fermentation process include but not limited to: first – the mixture of sugars produced in the lignocellulose hydrolysate (hexose: glucose, mannose, galactose, fructose; pentose: xylose, arabinose; others: lactose, sucrose)[13-14] require microorganisms that can selectively and simultaneously utilize at least the major sugars in the hydrolysate to produce the product of interest. Second, the typical concentrations of sugars produced in the hydrolysate may not be the optimum concentration to produce optimum product concentrations to minimize the downstream separation cost. For example, 1 – 23 g/L glucose was produced from 2 – 6% (w/v) sulfuric acid pretreated sugar cane bagasse[16], 1 – 20 g/L glucose from ionic liquid pretreated cellulose[17-18]) and about 20 g/L xylose and

58 g/L glucose produced from hydrogen peroxide-acetic acid pretreated Jerusalem artichoke hydrolyzed with enzymes (RUT-C30, pectinase and xylanase).[19] Additionally, the fermentation environment (substrate limitation and inhibition, product inhibition and potential recycle streams) affect the performance of the microorganisms. Lastly, the processing time raises questions as to the ideal length of time to run a batch fermentation process or when *in situ* product recovery, used to alleviate product inhibition, should be started. These challenges can potentially be addressed by Process Systems Engineering (PSE), the use of model-based methods and tools for the design, analysis, optimization, operation and control of complex chemical, biological or physical processes.[20-21]

Many of the performance metrics of fermentation processes that have to be simultaneously optimized in the face of limited resources (constraints) and process inputs (decision variables) are such that efforts to optimize one metric results in compromising or making one or more performance metric(s) simultaneously worse. This behavior is described as competing objectives. In the case of fermentation processes, the pairs of product selectivity and conversion, product quality and recovery cost, and profits and safety cost are often competing objectives in an optimization process. Single-objective optimization (SOO), the traditional optimization approach, either utilizes a weighted average sum of multiple objectives or designate one objective as the main objective function while treating other objectives as constraints. Conversely, a multi-objective optimization (MOO) approach simultaneously optimizes multiple objectives, even when they are competing. A SOO approach finds one unique optimal solution. If multiple optimal solutions exist (e.g. in the case of competing objectives), then different optimal solutions can be found by repeatedly solving the SOO problem with different initial guesses. MOO on the other hand is able to find Pareto-optimal solutions (a set of equally optimal solutions) and Pareto domain (a set of equally optimal decision variables), revealing the trade-offs that exist among, the often competing, objectives [27-28].

This work, therefore, uses a novel framework developed in our previous work in Chapters 3, 4 and 5 linking the optimization toolbox in MATLAB to the batch reactor in Aspen Plus coupled with a FORTRAN user kinetics subroutine. The data required to evaluate the objective function values are determined from Aspen Plus. The objective is to simultaneously maximize the ethanol yield, total ethanol produced, ethanol productivity (the total ethanol produced per batch fermentation time), and fraction of sugars converted (glucose and xylose) by optimizing the operating conditions (initial substrates concentrations and batch fermentation time) of a batch ethanolic process. A time-dependent fermentation mathematical model, developed by Leksawasdi et al.[137] based on recombinant *Z. mobilis* strain *ZM4* (*pZB5*) that utilizes a mixture of glucose and xylose simultaneously, is used as a model system to simulate the batch fermentation because ethanolic fermentation is susceptible to both substrate and product inhibitions. The results of the MOO approach using the genetic algorithm multi-objective optimization (*gamultiobj*) are compared to the traditional SOO approach using the genetic algorithm as a single-objective optimization (*ga*) in MATLAB. Additionally, the effect of product inhibition on the performance of the batch fermentation process is demonstrated by using an integrated batch and *in situ* product recovery process in the MOO and SOO.

6.3 Methods

6.3.1 Integrated MATLAB-Aspen Plus platform for optimization of batch ethanolic process based on time-dependent fermentation model

The batch reactor in the commercial process simulator, Aspen Plus, lacks the ability to incorporate the form of ordinary differential equations (ODEs) describing the batch ethanolic fermentation. In fact, the batch reactor in Aspen Plus can only incorporate power law reaction kinetics (an algebraic expression that relates the reaction rate to the reaction rate constants and a product of the concentration of components). The batch ethanolic fermentation model used as model system in this work uses ODEs that describe the reaction rates of each species in terms of constant reaction rate parameters and a complex dependence on concentration of components (using typical Monod-type of cell growth kinetics), and does not conform to the standard built-in power law model. Further, because Aspen Plus was traditionally designed for steady state refinery and chemical processes for which linear and shortcut optimization approaches are adequate, Aspen Plus can only handle SOO. Aspen Plus cannot solve the MOO problem that the batch ethanolic fermentation presents. To circumvent these challenges, a general framework developed in in Chapters 3, 4 and 5 that links a robust non-linear optimization solver in MATLAB (*gamultiobj* and *ga*) to the batch reactor in Aspen Plus coupled with a user FORTRAN kinetics subroutine (incorporates the non-standard batch ethanolic ODEs) through a component object module (COM) interface is used as shown in Figure 6.1. The COM interface establishes a two-way communication between MATLAB and Aspen Plus, allowing a circle of communication until the optimization is terminated when the stopping criteria is met. Details of the procedure to simulate the batch reactor as unsteady state process in Aspen Plus by linking the batch reactor to a FORTRAN user defined kinetics subroutine can be found in Chapter 4.

6.3.2 Prerequisites for process optimization of the batch ethanolic fermentation

6.3.2.1 Fermentation model used to simulate batch ethanolic fermentation

To simulate realistically the batch ethanolic fermentation, the following characteristics must be included in the fermentation model used: autocatalytic cell growth, microbial growth rate must be dependent on substrate concentration, cell growth mechanism must incorporate high substrate and product (ethanol) inhibitions and the rate of substrate utilization and product formation must be dependent on cell concentration. An unstructured kinetic model developed by Leksawasdi et al.[137] that simultaneously utilizes glucose and xylose based on a batch culture of the recombinant *Z. mobilis ZM4* (*pZB5*) was used to simulate the batch ethanolic fermentation process. The microbial kinetics were developed separately for glucose and xylose (eq L.1 and L.2 in Appendix L, respectively) and combined to describe the total microbial growth based on the two sugars using a relative weighing factor of glucose to xylose (sum equal to 1). Similar modeling equations were used for glucose and xylose utilization as well as ethanol production based on both glucose and xylose. The model incorporates kinetics for substrate limitation, substrate inhibition and product inhibition (ethanol threshold beyond which product inhibition is effective and maximum ethanol inhibition concentrations). The relative preferences of glucose to xylose uptake and utilization for autocatalytic cell production and ethanol production were taken as 65% of the maximum rate of glucose uptake and 35% of the maximum rate of xylose uptake, respectively for glucose and xylose initially present at a 1:1 mass ratio. The initial concentration of cells was assumed to be 0.003 g/L in all cases with no ethanol initially present.

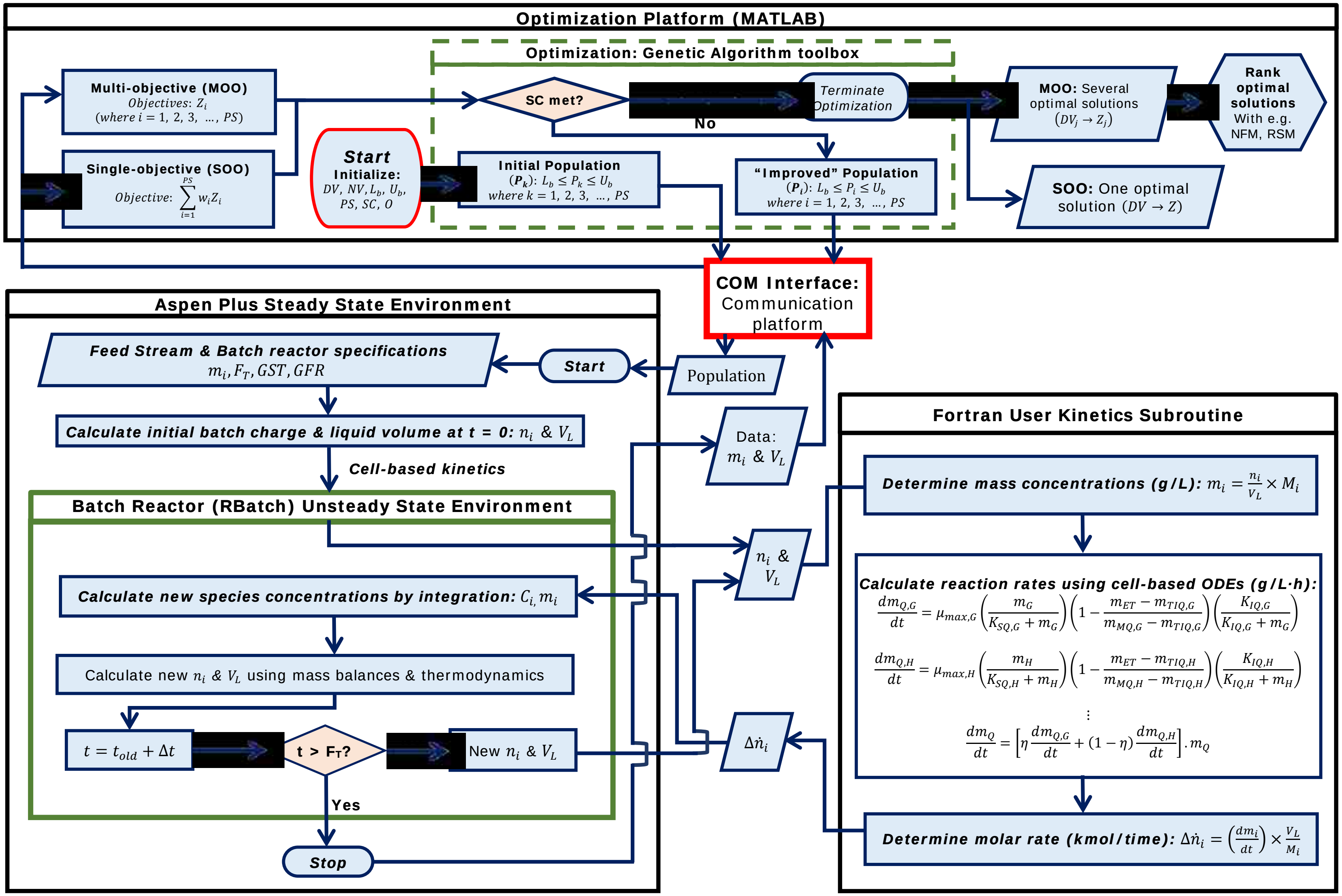

Optimization Platform (MATLAB)
Optimization: Genetic Algorithm toolbox
Multi-objective (MOO)
Objectives: Z_i
(where $i = 1, 2, 3, \ldots, PS$)
Single-objective (SOO)
Objective: $\sum_{i=1}^{PS} w_i Z_i$
Start
Initialize:
$DV, NV, L_b, U_b, PS, SC, O$
SC met?
No
Terminate Optimization
Initial Population
(P_k): $L_b \leq P_k \leq U_b$
where $k = 1, 2, 3, \ldots, PS$
"Improved" Population
(P_i): $L_b \leq P_i \leq U_b$
where $i = 1, 2, 3, \ldots, PS$
MOO: Several optimal solutions
$(DV_j \rightarrow Z_j)$
Rank optimal solutions
With e.g. NFM, RSM
SOO: One optimal solution $(DV \rightarrow Z)$
COM Interface:
Communication platform
Aspen Plus Steady State Environment
Feed Stream & Batch reactor specifications
m_i, F_T, GST, GFR
Start
Population
Calculate initial batch charge & liquid volume at t = 0: n_i & V_L
Cell-based kinetics
Data:
m_i & V_L
Batch Reactor (RBatch) Unsteady State Environment
n_i & V_L
Calculate new species concentrations by integration: C_i, m_i
Calculate new n_i & V_L using mass balances & thermodynamics
$t = t_{old} + \Delta t$
t > F_T?
New n_i & V_L
$\Delta \dot{n}_i$
Yes
Stop
Fortran User Kinetics Subroutine
Determine mass concentrations (g/L): $m_i = \frac{n_i}{V_L} \times M_i$
Calculate reaction rates using cell-based ODEs (g/L·h):
$\frac{dm_{Q,G}}{dt} = \mu_{max,G}\left(\frac{m_G}{K_{SQ,G} + m_G}\right)\left(1 - \frac{m_{ET} - m_{TIQ,G}}{m_{MQ,G} - m_{TIQ,G}}\right)\left(\frac{K_{IQ,G}}{K_{IQ,G} + m_G}\right)$
$\frac{dm_{Q,H}}{dt} = \mu_{max,H}\left(\frac{m_H}{K_{SQ,H} + m_H}\right)\left(1 - \frac{m_{ET} - m_{TIQ,H}}{m_{MQ,H} - m_{TIQ,H}}\right)\left(\frac{K_{IQ,H}}{K_{IQ,H} + m_H}\right)$
$\vdots$
$\frac{dm_Q}{dt} = \left[\eta \frac{dm_{Q,G}}{dt} + (1 - \eta)\frac{dm_{Q,H}}{dt}\right] . m_Q$
Determine molar rate (kmol/time): $\Delta \dot{n}_i = \left(\frac{dm_i}{dt}\right) \times \frac{V_L}{M_i}$

Figure 6.1 Framework of integrated MATLAB-Aspen Plus platform for optimization of batch ethanolic process based on time-dependent fermentation model
The notations are decision variables (*DV*), number of decision variables (*NV*), lower bounds of decision variables (L_b), upper bounds of decision variables (U_b), population size (*PS*), stopping criteria (*SC*), objective functions (*O*), component object module (COM), objective function values (*Z*), relative weight of each objective function (*w*), number of non-dominated solutions (*j*), population (*P*), *m* (mass concentration, g/L), F_T (total fermentation time, h), *GFR* (gas flow rate, L/min per L of broth), *GST* (time that gas stripping is initiated after batch fermentation, h), *n* (moles), V_L (liquid volume in reactor, L), *M* (molar mass, g/mol), *Q*, *G*, *H*, *ET* (cells, glucose, xylose, ethanol, respectively), *TIQ* (threshold inhibitory ethanol concentration in cell growth, g/L), *MQ* (maximum inhibitory ethanol concentration in cell growth, g/L), $\boldsymbol{\eta}$ (weighing factor for glucose utilization), $\Delta\dot{\boldsymbol{n}}$ (change in the molar rate, mol/time), K_{SQ}, K_{IQ}, (kinetic parameters), *C* (molar concentration, mol/L), *t* (current time), $\Delta\boldsymbol{t}$ (variable time step), t_{old} (previous time)

The details of the mathematical model and parameters are found eqs L.1 – L.8 in Appendix L. The ODEs were coded in a FORTRAN subroutine in Microsoft® Visual Studio 2013, compiled with Intel® FORTRAN Parallel Studio XE 2015 Composer Edition for Windows and saved in the same file as the Aspen Plus files, using Aspen Plus V9 (Aspen Technology, Inc., MA, USA). The laptop used was a Dell Precision M4800 with a 64-bit operating system, 16.0 GB RAM, Intel® core™ i7-4910MQ CPU @ 2.90 GHz and Windows 7 Professional.

6.3.3 Linking MATLAB and Aspen Plus for a two-way communication

MATLAB uses *actxserver* to create a Microsoft® Component Object Model (COM) automation server that can controls Aspen Plus. Through the COM interface, Aspen Plus can be opened, data written to and read, saved and closed (see details in Appendix M).

6.3.4 Batch ethanolic fermentation process optimization

After the initiation of the batch fermentation in Aspen Plus using the procedure outlined in our previous work in Chapter 4, the optimization process is carried out as shown in Figure 6.1. The genetic algorithm in MATLAB is linked with the batch reactor in Aspen Plus through the COM interface. Through the COM interface, the decision variables generated in the population (i.e. initial concentration of sugars and batch fermentation time) are written from MATLAB to the batch reactor in Aspen Plus. After running the batch reactor in Aspen Plus (for the fermentation time and substrates concentrations), the concentrations of glucose, xylose, ethanol and volume of the liquid contents in the fermentor are read from the batch reactor through the COM interface to MATLAB, to evaluate the objective function values. The batch reactor is coupled with a FORTRAN user kinetics subroutine to simulate the batch fermentation as an unsteady state process in Aspen Plus.

The genetic algorithm toolbox in MATLAB minimizes a single objective using the *ga* solver or multiple objective functions simultaneously using the *gamultiobj* solver.

The objectives of the MOO was to simultaneously maximize the ethanol yield (mass of ethanol produced per mass of sugars utilized, eq 6.1), total ethanol produced (eq 6.2), ethanol productivity (total ethanol produced per batch fermentation time, (eq 6.3)) and total fraction of sugars converted (ratio of the total mass of sugars utilized to the total mass of sugars initially present, (eq 6.4)) subject to the constraints: the initial concentration of substrates (eq 6.5) and batch fermentation time (eq 6.6).

MOO problem formulation

Maximize:

Ethanol yield, g-ethanol/g-sugars utilized = $[m_{ET,final}V_{L,final}]/[(m_{G,initial}V_{L,initial} - m_{G,final}V_{L,final}) + m_{H,initial}V_{L,initial} - m_{H,final}V_{L,final})]$ (6.1)

Ethanol produced, g/L $= m_{ET,final}$ (6.2)

Ethanol productivity, g/L.h $= m_{ET,final}/F_T$ (6.3)

Fraction of sugars converted $= \left[(m_{G,initial}V_{L,initial} - m_{G,final}V_{L,final}) + (m_{H,initial}V_{L,initial} - m_{H,final}V_{L,final})\right]/(m_{G,initial}V_{L,initial} + m_{H,initial}V_{L,initial})$ (6.4)

Subject to the constraints:

Initial substrates concentration (ISC): $10 \leq ISC \leq 300\ g/L$ (6.5)

Batch fermentation time (F_T): $5 \leq F_T \leq 100\ h$ (6.6)

where m_G, m_{ET}, m_H, V_L, and F_T are mass concentrations (g/L) of glucose, ethanol, and xylose, liquid volume of the contents in reactor, respectively. “*initial*” and “*final*” are used to denote parameters at the beginning of fermentation and at the end of fermentation, respectively.

The objective of the SOO (eq 6.7) was to maximize the weighted sum of the objectives in eqs 6.1 – 6.4 subject to the same constraints as the MOO in eqs 5 and 6. Equal relative weights were used for each objective.

SOO problem formulation

Maximize: $\sum_{i=1}^{N} Z_i w_i$ (6.7)

where N, Z, w are number of objectives, objective function values (ethanol yield, total ethanol produced, ethanol productivity and fraction of sugars converted) and relative weights (0.25) of each objective function value, respectively.

6.3.4.1 Initialization of optimization and generation of initial population (*Pk*)

Both the MOO and SOO use the same decision variables, number of variables, upper and lower bounds, population size and stopping criteria but are run separately. The difference between the MOO and SOO optimization process is in the formulation of the objective functions as shown in section 6.3.4. The MOO and SOO uses the *gamultiobj* and *ga* solvers, respectively, with the default parameter settings in MATLAB unless otherwise stated. The optimization process is initialized by specifying : the decision variables (DV = ISC, F_T), the number of decision variables (NV = 2), lower (L_b = 10 g/L & 5 h) and upper (U_b = 300 g/L & 100 h) bounds, population size (PS = 1000), stopping criteria (SC = 0.01, function tolerance, representing the smallest uncertainty in the measurement of a typical ethanolic batch fermentation parameters[1]) and the objective function expressions (O). A set of pairs (ISC and F_T) of random points are generated as the initial population (P_k) with size equal to PS between the range of decision variables specified through L_b and U_b.

6.3.4.2 Determination of objective function values

For the elements in the population generated (in this case 1000 sets), the objective function values must be determined for each set. The objective function values are then used to test if the stopping criteria is met. In order to determine the objective function values for each pair of initial sugars concentration and batch fermentation time, the initial sugars concentration and fermentation time are first written to Aspen Plus. For example, initial sugars of concentration of 100 g/L and fermentation time of 50 h are written to Aspen Plus by opening the batch reactor file initially created through the COM interface. The initial concentration of glucose/xylose are both changed to 100 g/L and the fermentation time for the batch reactor is changed to 50 h in Aspen Plus. The batch reactor is then reinitialized, run and saved.

To run the batch reactor as shown in Figure 6.1, the molar amount of each component (n_i, where i = glucose, xylose, cells, ethanol and water) and the liquid volume of the contents in the reactor are accessed in the FORTRAN user kinetics subroutine to evaluate the mass concentration of each component (m_i). The calculated concentrations are then used in the biological ODEs describing the batch ethanolic fermentation to determine the rates of consumption or generation of each component in the batch process. The change in the molar rate of each component ($\Delta\dot{n}$) for the current integration time are passed from the subroutine to the batch reactor in Aspen Plus. In the batch reactor, new species concentrations are calculated by integration (using the default variable-step-size Gear's algorithm). The new state of the batch reactor is evaluated using material and energy balances as well as phase equilibria equations and thermodynamic models. If the current integration time (t) is less than the specified fermentation time (F_T), the new state (n_i and V_L) are passed to the subroutine for further integration until the stopping criteria is met (batch fermentation time). The simulation is then saved, the data required to evaluate the objective function values (mass concentrations of components, liquid volume) for the pair of initial sugars concentration and batch fermentation time are passed from Aspen Plus to MATLAB through the COM interface. All the elements in the population (in this case 1000) are passed from MATLAB to the batch reactor through the COM interface and the data required to evaluate the objective function values are read from the batch reactor to MATLAB until the complete set of objective function values is obtained.

6.3.4.3 Optimization in MATLAB

With the set of objective function values equal to the population size (in this case 1000 sets), the genetic algorithm determines if the stopping criteria (the change is the spread of the Pareto front over successive generations is less than the function tolerance specified, 0.01) is met. For a pair of objective function values, if at least one objective function value of solution 1 is better than solution 2, then solution 1 is described as non-dominated with respect to solution 2.[138] For each generation (the set of 1000 objective function values in this case) a set of non-dominated solutions are selected from the total population and the set of non-dominated solutions are updated after every generation. A new set of 1000 "improved" population, representing the next generation, are generated and passed to Aspen Plus through the COM interface to determine the corresponding objective function values. In generating the "improved" population, some of the individuals in the current generation with the best scores (the objective function values converted into scores) are selected as the elite to be part of the population in the next generation. The remaining number of individuals needed to form the total population, 1000 in this case, are generated using parents selected from the current generation to produce children for the next generation. Children are produced from selected parents through crossover – combing the elements from different parents and mutation – making random changes to the elements of a single parent.[139] Through successive generations, the stopping criteria is met and optimization is terminated.

6.3.4.4 Analyses of optimization results: MOO versus SOO

At the end MOO process, a set of decision variables (DV_j) and the corresponding objective function values (Z_j), where j is the number of non-dominated solutions, are selected from the final population. The number of non-dominated solutions (Pareto domain) obtained in the MOO can be ranked using ranking procedures such as the Net Flow Method (NFM) and Rough Set Method.[138] The NFM uses four sets of ranking parameters to score and rank the non-dominated solutions; indifference, preference and veto thresholds, and relative weight of each objective function. The indifference threshold represents a value below which if the difference between two objective function values falls, the two objective function values cannot be distinguished. This means, one objective function value cannot be chosen over another if the difference between the objective function values is below the indifference threshold. If the difference between a pair of objective function values is greater than the preference threshold, the better solution is chosen. For example, for a maximization problem, the preferred value will be the objective function with the larger value and vice versa. When the difference between two objective function values is greater than the veto threshold, one solution is banned against the other solution. The indifference, preference and veto thresholds are established such that 0 ≤ indifference threshold ≤ preference threshold ≤ veto threshold. The details of the NFM method from the work of Jules Thibault[138] are presented in Appendix K. In this work, equal relative weights were set for the objective function and the indifference, preference and veto thresholds set at 5%, 10% and 30% of the range of each objective function values in the Pareto domain, respectively. These thresholds were chosen using the thresholds in the work of Aida et al. [84] as a guide. At the end of the SOO, one optimal solution, a decision variable (DV) and the corresponding weighted sum of objective function value (Z), is obtained. The decision variable obtained in the SOO can then be used to evaluate the objective function values.

6.3.5 Optimization of integrated batch ethanolic fermentation and *in situ* product recovery

To demonstrate how the performance of a batch fermentation is enhanced when *in situ* product recovery is integrated with the batch fermentation, the integrated batch ethanolic fermentation and *in situ* gas stripping process is used in the optimization process. The details of how an integrated batch fermentation (based on time-dependent model) with *in situ* gas stripping can be simulated is found in our previous work in Chapter 4. In summary, the batch fermentation is simulated as unsteady state process by linking the batch reactor in Aspen Plus to a FORTRAN user kinetics subroutine (evaluate the rates of consumption or generation of each component). A gas stream (nitrogen) is fed continuously to the unsteady state batch reactor to simulate an integrated batch fermentation with *in situ* product recovery by gas stripping. The optimization procedure for the integrated batch ethanolic fermentation and *in situ* gas stripping is the same as described in sections 6.3.4.1, 6.3.4.2, 6.3.4.3 and 6.3.4.4.

The objectives of the MOO was to simultaneously maximize the ethanol yield (eq 6.8), total ethanol produced (eq 6.9), ethanol productivity (eq 6.10) and total fraction of sugars converted (eq 6.11) subject to the constraints: the initial concentration of substrates (eq 6.12), batch fermentation time (eq 6.13), the gas flow rate of N_2 used in gas stripping (eq 6.14) and the gas stripping initiation times (eq 6.15). In addition, the gas stripping initiation times should always be less than or equal to the batch fermentation time (equation 16). The concentrations of ethanol in the reactor and the stripped stream are both accounted for when determining the total ethanol produced.

MOO problem formulation

Maximize:

Ethanol yield, g-ethanol/g-sugars utilized = $[m_{ET,reactor,final}V_{L,reactor,final} + m_{ET,stripped\ stream,final}V_{L,stripped\ stream,final}]/[(m_{G,initial}V_{L,initial} - m_{G,final}V_{L,final}) + m_{H,initial}V_{L,initial} - m_{H,final}V_{L,final})]$ (6.8)

Ethanol produced $= [m_{ET,reactor,final} + m_{ET,stripped\ stream,final}]F_T$ (6.9)

Ethanol productivity $= [m_{ET,reactor,final} + m_{ET,stripped\ stream,final}]F_T$ (6.10)

Fraction of sugars converted $= (m_{G,initial}V_{L,initial} - m_{G,final}V_{L,final}) +$
$(m_{H,initial}V_{L,initial} - m_{H,final}V_{L,final})]/(m_{G,initial}V_{L,initial} + m_{H,initial}V_{L,initial})$ (6.11)

MOO subject to the following constraints:

Initial substrates concentration (ISC): $10 \leq ISC \leq 300\ g/L$ (6.12)

Batch fermentation time (F_T): $5 \leq F_T \leq 100\ h$ (6.13)

Gas stripping start time (*GST*): $5 \leq GST \leq 100\ h$ (6.14)

Gas flow rate (GFR): $0.1 \leq GFR \leq 3$ L/min per L of broth (6.15)

Gas stripping start time is less or equal to the batch fermentation time: $GST \leq F_T$ (6.16)
where m_G, m_{ET}, m_H, V_L, and F_T, "*initial*" and "*final*" have the same meanings as defined in eqs 6.1 – 6.6 in section 6.3.4.

The objective of the SOO presented in equation 6.17 was to maximize the weighted sum of the objectives in eqs 6.8 – 6.11 (using equal weights for each objective) subject to the same constraints as the MOO (eqs 6.12 – 6.16).

SOO problem formulation

Maximize $\sum_{i=1}^{N} Z_i w_i$ (6.17)

where N, *Z*, *w* have the same meanings as defined in equation 7 in section 2.2.

See sample MOO and SOO MATLAB codes in Appendix P.

6.4 Results and Discussion

6.4.1 Optimization of batch ethanolic fermentation using MOO and SOO

Figure 6.2 shows the Pareto domain of the MOO maximizing the ethanol yield, ethanol productivity, total ethanol produced, and fraction of sugars converted (glucose/xylose) simultaneously by manipulating the batch fermentation time (5 – 100 h) and initial substrates concentration (10 – 300 g/L). Equal relative weights were used for each objective to score and rank the Pareto-optimal solutions using the Net Flow Method (NFM). The decision variables (initial concentration of sugars and the batch fermentation time) and the corresponding objective function values with the highest score from the NFM is selected as the best optimal point among the sets of equally optimal solutions.

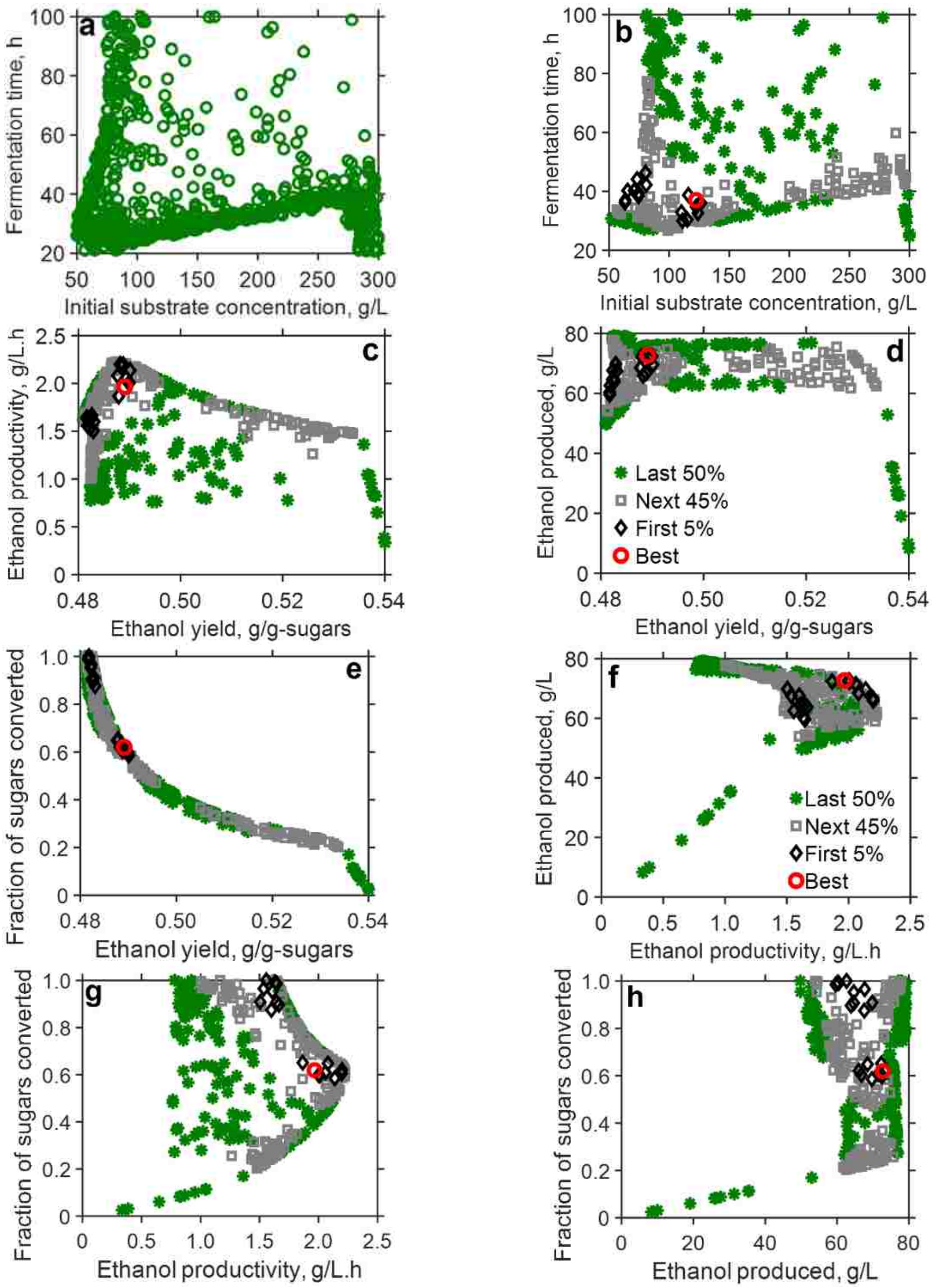


Figure 6.2 Plot of the MOO results for batch ethanolic fermentation

The notations are: a) final population b) decision variables c) ethanol productivity versus ethanol yield d) total ethanol produced versus ethanol yield e) fraction of sugars converted versus ethanol yield f) total ethanol produced versus ethanol productivity g) fraction of sugars converted versus ethanol productivity h) fraction of sugars converted versus total ethanol produced. The red circle, black diamonds, grey squares and green asterisk represents the best, first 5 %, next 45 % and the last 50 %, respectively, ranked using the NFM method

The ranked solutions are grouped into zones representing the best (red circle), first 5% (black diamonds), next 45% (gray squares) and last 50% (green asterisks). Figure 6.2a showing the final population represents a large area of the decision space, where "weaker" performing points have been eliminated in successive generations in the genetic algorithm process. 350 non-dominated points were chosen from the final population of 1000 points as the decision variables as shown in Figure 6.2b. The NFM found a compromised best decision variable of 122.4 g/L initial concentration of sugars and fermentation time of 37 h. This pair of decision variables corresponds to 0.489 g/g-sugars, 1.97 g/L.h, 72.7 g/L and 0.62 ethanol yield, ethanol productivity, total ethanol produced and fraction of sugars converted, respectively. The trade-off between the competing pair of objectives, ethanol yield versus fraction of sugars converted (Figure 6.2e) and ethanol productivity versus ethanol yield (Figure 6.2c) can clearly been seen. For example, lower ethanol yield resulted in a higher ethanol productivity and vice versa.

Majority of the optimal solutions predicted for the total ethanol produced lies between 60 and 79 g/L, which is in the neighborhood of the maximum ethanol concentration beyond which ethanol production ceases of about 75.4 g/L and 81.2 g/L for glucose and xylose fermentation, respectively for *Z. mobilis*[137] (Figure 6.2d, Figure 6.2f, and Figure 6.2h). The theoretical maximum yield for ethanol production is 0.511 g ethanol/g sugars based on hexoses and pentoses.[140] The range of ethanol yield is from 0.48 to 0.54 as shown in Figures 6.2c, 6.2d, and 6.2e, indicting some of the optimal solutions in the MOO are greater than the theoretical maximum yield of 0.511. The optimal solutions with ethanol yield greater than 0.511 generally corresponds to very low sugars utilization (fraction of sugars converted is less than 0.3). These solutions, therefore, may be practically infeasible.

When the best optimal and first 5% ranked solutions are considered together, the MOO process predicted two regions, (63 – 75 g/L initial concentration of sugars and 37 – 44 h fermentation time) and (122 – 126 g/L initial concentration of sugars and 34 – 37 h fermentation time), as the decision variables. These two regions of decision variables may offer flexibility to the fermentation design engineer in designing batch fermentation processes. The zone represented by the first 5% optimal points in the MOO could represent situations where perturbations or deviations in process inputs and conditions affect the performance of the batch ethanol process.

On the other hand, the SOO (using equations 6.5 – 6.7) predicted an initial concentration of sugars of 84.5 g/L and a batch fermentation time of 99. 6 h, resulting in an ethanol yield of 0.482 g/g-sugars, 79.4 g/L total ethanol produced, 0.797 g/L.h ethanol productivity and 0.98 fraction of sugars converted. The SOO approach offers only one optimal solution without alternative design variables. The typical concentrations of sugars obtained after pretreatment and hydrolysis of lignocellulosic biomass is low (e.g., 1 – 23 g/L glucose[16]). Both the MOO and SOO predict an optimal initial concentration of sugars of 122.4 g/L and 84.5 g/L, respectively, which are relatively higher compared to the typical concentration of sugars in lignocellulosic hydrolysate. This suggests the typical concentration of sugars obtained in lignocellulosic hydrolysis may have to be concentrated prior to fermentation for optimal performance of the batch fermentation process.

The time-dependent concentrations (for cells, glucose, xylose and ethanol in the fermentor) plotted from the best optimal point of 122.4 g/L (initial concentrations of glucose and xylose) and 37 h of fermentation, identified from the MOO of a batch fermentation without *in situ* separation (blue line) is shown in Figure 6.3. The cell growth kinetics (Figure 6.3a) shows cell death occurring after about 30 h of batch fermentation, corresponding to an ethanol concentration of 65.9 g/L (Figure 6.3d). 65.9 g/L of ethanol is greater than the maximum inhibitory ethanol concentrations for cell growth based on glucose and xylose of 57.2 g/L and 56.3 g/L, respectively for *Z. mobilis.*[137] The residual sugars are relatively high (for xylose), 1.2 g/L glucose and 84.1 g/L xylose, resulting in 0.62 fraction of sugars converted at the end of the 37 h of batch fermentation. This observation emphasizes the crippling effect that product (ethanol) inhibition has on microbial cell growth and consequently the performance (yields, titers, productivities, conversion) of the batch ethanolic fermentation.

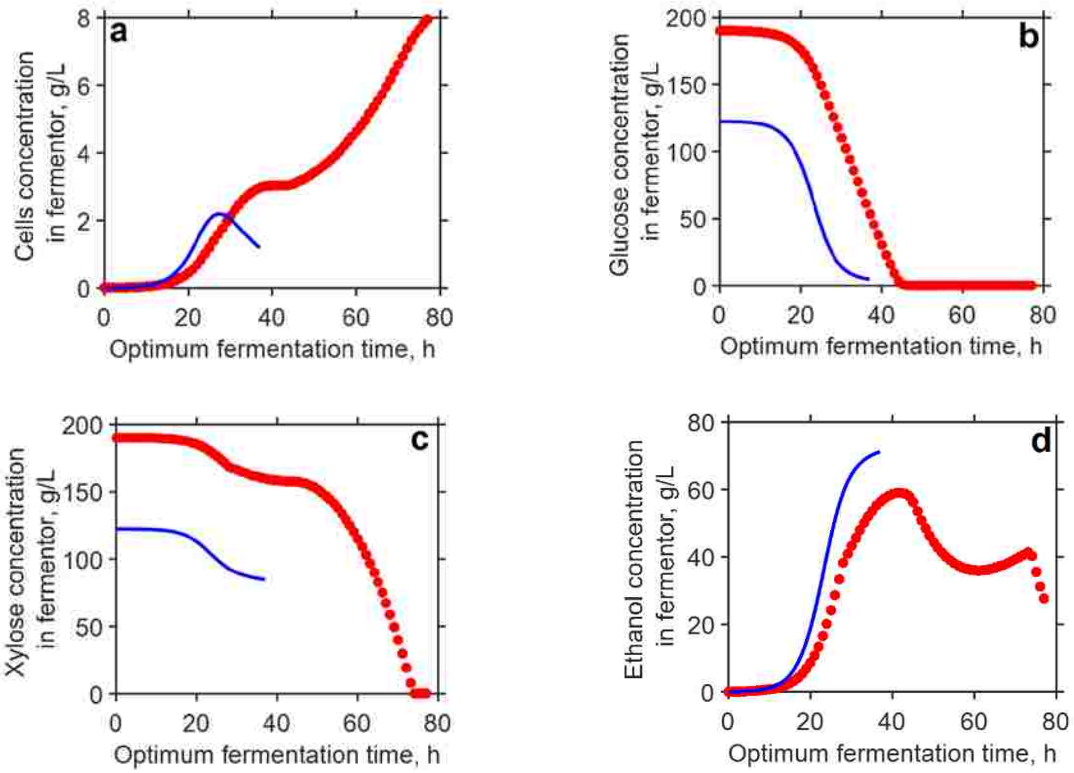


Figure 6.3 Concentration profile of the contents of the fermentor for the best optimal solution in the MOO (ranked with NFM) of the batch ethanolic fermentation (blue line, —) and integrated batch ethanolic fermentation with *in situ* separations (red filled circle, ●) The notation is a) cells concentration b) glucose concentration c) xylose concentration d) ethanol concentration

6.4.2 Optimization of integrated batch ethanolic fermentation with *in situ* separation using MOO and SOO

To demonstrate how the performance of the batch ethanolic fermentation is improved when the batch process is integrated with an *in situ* product recovery, an integrated batch and *in situ* gas stripping process was used in both the MOO and SOO. The objective was to maximize the ethanol yield, ethanol productivity, total ethanol produced and the fraction of sugars converted by manipulating the initial substrates concentration, batch fermentation time, gas stripping initiation times and the gas flow rate employed (equations 6.8 – 6.17). Figure 6.4 shows the decision variables in the MOO and the resulting objective function values for the MOO ranked with the NFM for the integrated batch and *in situ* gas stripping process.

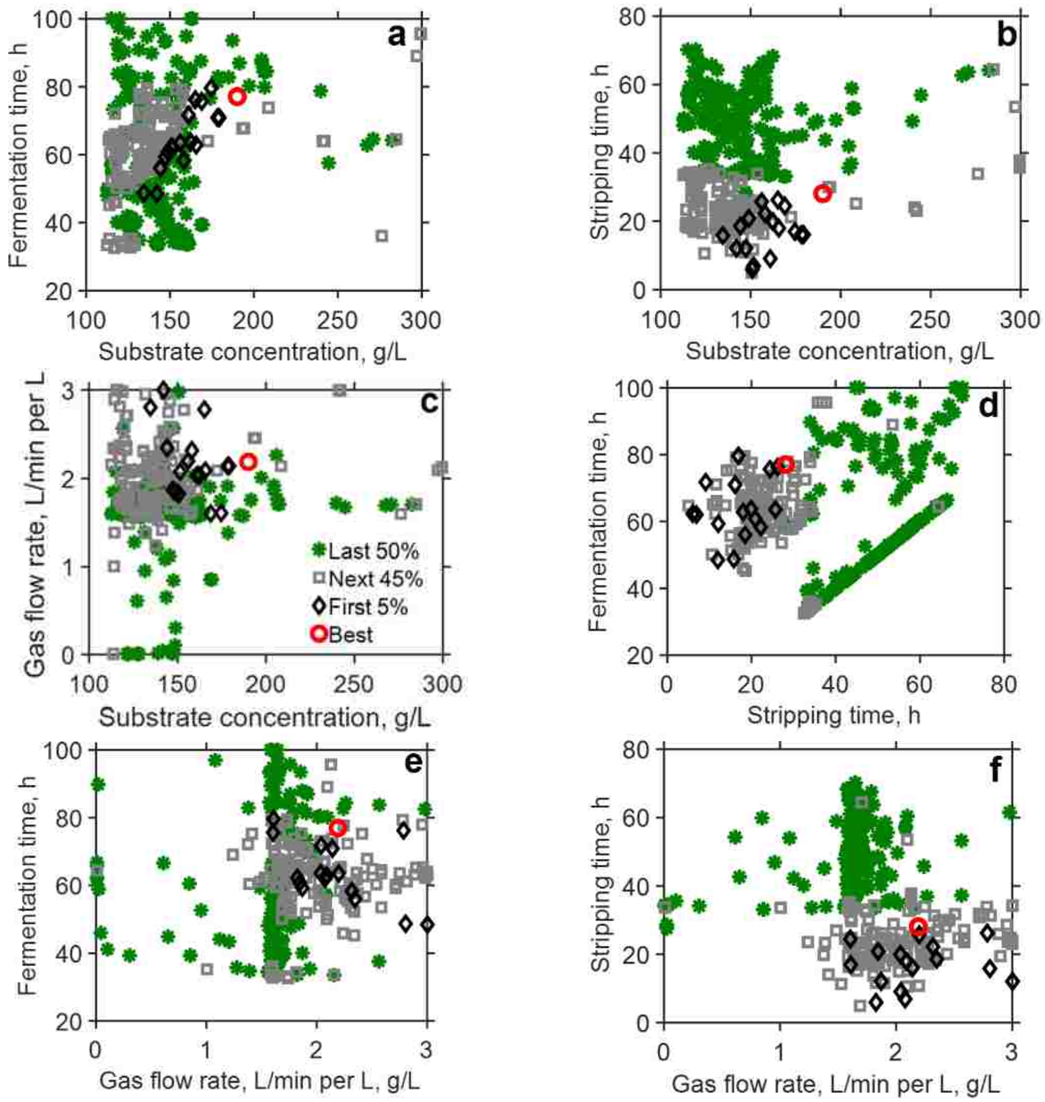


Figure 6.4 Plot of the decision variables (ranked with NFM) in the MOO results for integrated batch ethanolic fermentation and *in situ* separation
The notation is a) batch fermentation time versus initial substrates concentration b) gas stripping initiation time versus initial substrates concentration c) gas flow rate versus initial substrates concentration d) batch fermentation time versus gas stripping initiation time e) batch fermentation time versus gas flow rate f) gas stripping initiation time versus gas flow rate. The red circle, black diamonds, grey squares and green asterisk represents the best (top ranked), first 5 %, next 45 % and the last 50 %, respectively, ranked using the NFM method

When product inhibition is alleviated by removing ethanol as it is produced via gas stripping, an optimal best initial concentration of sugars of 190.1 g/L, 77 h fermentation time, 28.1 gas stripping initiation times and 2.2 L/min per L of broth gas flow rate is predicted for the MOO. These MOO decision variables resulted in an ethanol yield of

0.478 g/g-sugars, ethanol productivity of 2.29 g/L.h, total ethanol produced of 176.5 g/L and fraction of sugars converted of 1.0. The SOO predicted initial sugars concentration of 254.6 g/L, a batch fermentation time of 84.9 h, gas stripping initiation time of 15.3 h and gas flow rate 0f 2.3 L/min per L of broth for the decision variables in the integrated batch fermentation and *in situ* product recovery process. These SOO decision variables represent an ethanol yield of 0.450, ethanol productivity of 2.50 g/L.h, total ethanol produced of 210.2 g/L and fraction of sugars converted of 0.98. The SOO optimization predicted higher initial concentrations of sugars (254.6 versus 190.1 g/L), total ethanol produced (210.2 versus 176.5 g/L) and ethanol productivity (2.50 versus 2.29 g/L.h) compared to the MOO, respectively, for the best optimal solution. However, the predicted ethanol yield in the SOO approach was less than that of the MOO (0.450 versus 0.478, respectively).

As shown in Figure 6.3d (filled red circle), when ethanol is removed *in situ* after 28 h of batch fermentation, the concentration of ethanol in the fermentor is always below 59 g/L (maximum ethanol concentration reached in the reactor). 59 g/L is slightly higher than the maximum ethanol concentration beyond which cells growth ceases for the model system used (57.2 g/L and 56.3 g/L for cell growth based on glucose and xylose, respectively[137]) but lower than the maximum ethanol concentration above which glucose and xylose uptake for ethanol production is inhibited (75.4 and 81.2 g/L, respectively[137]). As a result, the cells are able to utilize the remaining sugars (mostly xylose – about 168 g/L left at 28 h of fermentation) to produce more ethanol. There is a significant change in the cell growth kinetics (Figure 6.3a), where in the batch ethanolic fermentation alone, cell death occurs as a result of ethanol inhibition (~65.9 g/L of ethanol after 30 h of batch fermentation). In the case of the integrated batch and *in situ* separations, the cells are able to resuscitate after a brief period of constant growth between 38 – 46 h (cells concentration remains approximately 3 g/L) until the ethanol concentration falls below the maximum ethanol inhibitory concentrations of 57.2 and 56.2 g/L, for glucose and xylose respectively, for cell growth.

As previously discussed, MOO of the integrated batch ethanolic fermentation and *in situ* product separation predicted an optimal best initial sugars concentration of 190.1 g/L, 77 h fermentation time, 28.1 gas stripping initiation times and 2.2 L/min per L of broth gas flow rate. As shown in Table 1, these decision variables resulted in an ethanol yield of

0.478 g/g-sugars, ethanol productivity of 2.29 g/L.h, total ethanol produced of 176.5 g/L and fraction of sugars converted of 1.0. The initial sugars concentration predicted is increased by 55% in the integrated batch and *in situ* product recovery relative to the batch fermentation process alone for the best optimal solution in the MOO. However, the ethanol yield reduced by 2.3% with a 16% enhancement in the ethanol productivity, 143% increase in the total ethanol produced and an improvement in the fraction of sugars converted of 62% when the integrated batch with *in situ* separations is compared with the batch fermentation process.

Table 6.1 Comparison of the predicted decision variables and corresponding objective function values of the best optimal solution (ranked with NFM) in the MOO for the batch ethanolic fermentation without and with in situ separations

Parameter	Batch fermentation without separations	Integrated batch with *in situ* separations	Enhancement, %
	Decision variables		
Initial substrate concentration, g/L	122.4	190.1	+55
Fermentation time, h	37	77	+108
Gas stripping initiation time, h	–	28.1	
Gas flow rate, L/min per L	–	2.2	
	Objective function values		
Ethanol yield, g/g	0.489	0.478	-2.3
Ethanol productivity, g/L.h	1.97	2.29	+16
Total ethanol produced, g/L	72.7	176.5	+143
Fraction of sugars converted	0.62	1.0	+62

It can be inferred in general that a more realistic range of ethanol yields is obtained in the integrated batch and *in situ* separations process (Figure 6.5g, Figure 6.5h and Figure 6.5i), where most of the ethanol yield values are below the theoretical maximum ethanol yield of about 0.511 g/g-sugars.[140] Two distinct regions of non-dominated optimal solutions are generated for the plot of ethanol produced versus ethanol yield (Figure 6.5h), ethanol produced versus ethanol productivity (Figure 6.5j), fraction of sugars converted versus ethanol productivity (Figure 6.5k), and fraction of sugars converted versus ethanol productivity (Figure 6.5l).

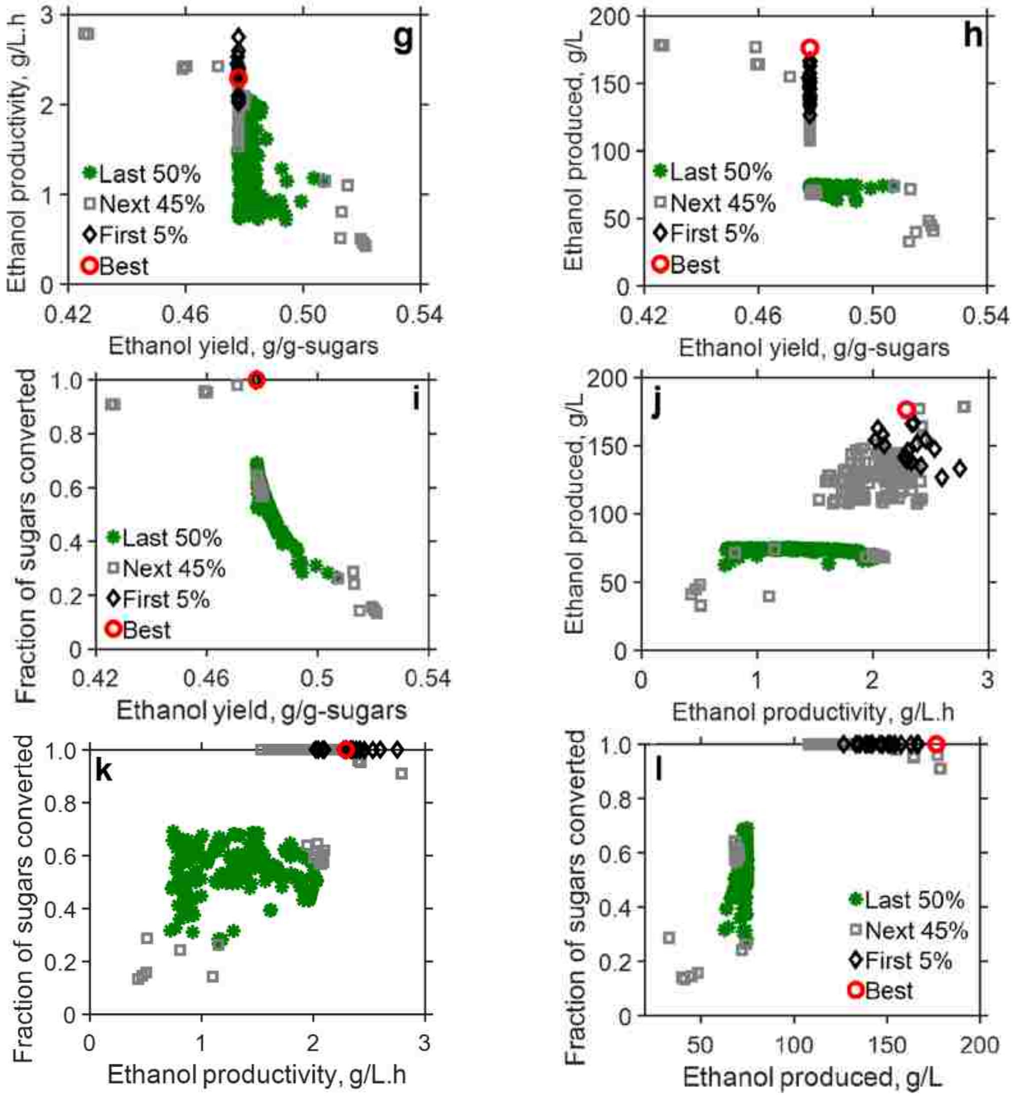


Figure 6.5 Plot of the decision variables (ranked with NFM) in the MOO results for integrated batch ethanolic fermentation and *in situ* separation.
The notation is g) ethanol productivity versus ethanol yield h) total ethanol produced versus ethanol yield i) fraction of sugars converted versus ethanol yield j) total ethanol produced versus ethanol productivity k) fraction of sugars converted versus ethanol productivity l) fraction of sugars converted versus total ethanol produced. The red circle, black diamonds, grey squares and green asterisk represents the best (top ranked), first 5 %, next 45 % and the last 50 %, respectively

For example, two distinct regions in the ranges of 0.13 – 0.69 and 0.95 to 1.0 (fraction of sugars converted), 0.43 – 0.478 and 0.478 – 0.521 g/g (ethanol yield), 32.9 – 74.5 and 107.5 – 178.6 g/L (total ethanol produced). This observation may suggest that there exist multiple optimal regions (local and global solutions) that MOO helps to reveal.

For example, the range of 32.9 to 74.5 g/L ethanol produced and 0.478 – 0.521 ethanol yield correspond to the MOO results for the batch ethanolic fermentation alone.

6.5 Conclusion

In this work, the traditional and commercial Process Systems Engineering tools (Aspen Plus and MATLAB) and methods were integrated on a novel framework. An unsteady state batch fermentation in Aspen Plus was linked to the robust optimization in MATLAB through a COM interface to maximize the process inputs and operating conditions of a batch ethanolic fermentation process. Both MOO and SOO approaches demonstrated the typical sugars concentration obtained in lignocellulosic hydrolysate may have to be concentrated (by an order of magnitude higher) to optimize sugars utilization and ethanol production in the batch fermentation process. By integrating the batch fermentation with an *in situ* product recovery, the improvement in the performance of the fermentation process was significant; for example, the best optimal solution obtained in the MOO was shown to use 55% more concentrated initial sugars to improve the ethanol productivity, total ethanol produced and fraction of sugars converted by 16, 143, and 62 %, respectively, relative to the batch fermentation without *in situ* product recovery. The MOO revealed the trade-offs and interactions that may exist among fermentation process parameters. Additionally, the MOO approach presented many equally optimal alternative solutions that can be used to support the understanding of batch fermentation dynamics and what-if-analysis scenarios to mirror how deviations in process inputs and conditions may affect the performance of the batch process. This work demonstrates how process insights can be gained using virtual experimentation through *in silico* analyses for processes where laboratory experiments alone may be inadequate, time consuming or too costly. Through such a platform, the fermentation experimentalist has a decision-support tool to guide the choice of design variables and conditions for optimum process performance.

Chapter 7

Conclusion and Future Directions

7.1 Conclusion

Chapters 3 – 6 discussed the challenges, proposed and applied a framework to solve some of the challenges, and prospects for applying Process Systems Engineering tools and Methods to fermentation-based biorefineries. Because traditional process simulators were principally developed for the chemical and refinery industries, these PSE tools partially or fully lack built-in unit models with features that can readily simulate fermentation processes and optimization solvers that can handle the inherently non-linear MOO problem that are often encountered in fermentation-based biorefineries. In Chapter 3, the absence of typical key fermentation components, such as cells, in the database, the ability to simulate the inherently unsteady state fermentation processes and the absence of robust optimization tools in traditional PSE process simulators were identified as some of the major challenges hindering the application of PSE tools to fermentation-based biorefineries. A simple but effective technique, component substitution – the use of all the known physical properties of a databank component for the unknown physical properties of a non-databank component, was shown to be effective in representing non-databank components.

A novel integrated platform that links a traditional process simulator with an external robust optimization solver through a communication platform that allows data exchange between the process simulator and optimization solver was proposed in Chapter 3. To develop the proposed framework, an unsteady batch fermentation without and with *in situ* product recovery was developed in Chapter 4 using the commercial steady state process simulator, Aspen Plus. The time-dependent fermentation model describing the fermentation process in the form of ODEs were incorporated in a FORTRAN user kinetics subroutine because the form the ODEs does not conform to the built-in power law reaction kinetics in Aspen Plus. The unsteady state batch fermentation was validated when the simulation results match the integration of the ODEs in MATLAB quantitatively. Further, the trends in the results of the unsteady state batch fermentation are shown to be consistent with experimental trends in literature. This is an important validation where simulation results are reconciled with literature results to test the robustness of the developed procedure and the ability of the thermodynamic models to give accurate predictions that match experimental trends.

Unlike previous separate steady state fermentation and steady state separations (gas stripping), the product profile is demonstrated to be sensitive to the gas flow rate employed, demonstrating the importance of linking a time-dependent fermentation model with *in situ* separations for the systematic analyses of fermentation processes. The competing nature of the ABE selectivity and the concentration of ABE in the stripped stream is clearly observed, where a lower gas flow rate results in a lower selectivity of ABE and a high concentration of ABE in the condensate and vice versa. The techniques developed in Chapter 3 and the results are powerful and important developments that set the ground work for extension of unsteady state batch fermentation simulations to other fermentation schemes such as fed-batch and continuous fermentation without and with *in situ* product recovery to help guide the fermentation experimentalist.

The utility of the proposed framework in Chapter 3 was demonstrated in Chapter 5 where an existing commercial process simulator, Aspen Plus, was linked to the genetic algorithm multi-objective and single-objection optimizations in MATLAB through a COM interface, providing a two-way communication for data exchange. The MOO approach is demonstrated to be a global optimization approach in which the decision variables span the entire design space (search region). The best optimal solution selected from the equally optimal solutions in MOO scored and ranked with the NFM was shown to be consistent the operating conditions frequently used in laboratory experiments. The ability of the MOO approach to shed light on the effect of process conditions and inputs on the performance of the integrated fermentation and *in situ* separation process to support missing existing literature data was presented. For instance, there is very limited reporting on the effect of different gas flow rates on the selectivity of ABE in literature. Thus, the insights that virtual experimentation brings to the table when laboratory data is inadequate is brought to bear.

The sheer volume of variables, complexity and interaction of different decision variables that are faced in fermentation-based biorefineries require the use of PSE tools through *in silico* analyses to help choose optimum decision variables for process design and optimization for optimum process performance. In Chapter 6, the novel platform proposed in Chapter 3 was applied to optimize the operating conditions of a batch ethanolic fermentation (batch fermentation time and initial concentration of sugars) to maximize the total ethanol produced, ethanol productivity, ethanol yield and the fraction of sugars converted. A time-dependent fermentation model system based on the consumption of glucose and xylose was used in an unsteady state batch fermentation in Aspen Plus (using techniques developed in Chapter 4) linked to the robust optimization in MATLAB through a COM interface. The major contributions of this work are that prior concentration of the sugars obtained in lignocellulosic hydrolysate (at least by an order of magnitude) are required to optimize substrate utilization and alcohol production in batch fermentations as demonstrated by the MOO and SOO approaches. Further, with a higher initial substrate concentration utilized in the batch fermentation process, it is imperative to integrate the batch fermentation with *in situ* process recovery to alleviate product inhibition, improving the performance of the fermentation process significantly. It is demonstrated that the best optimal solution obtained from scoring and ranking the equally optimal solutions from the MOO uses 55% more concentrated initial substrate and enhanced the productivity of ethanol, total ethanol produced an fraction of sugars converted by 16, 143, and 62%, respectively when the integrated batch with in situ product recovery was compared to the batch process alone. Thus, this work demonstrated process insights that are ordinarily hidden or too difficult to investigate through laboratory experiments. Additionally, as a decision-support tool, the platform developed can be used to guide the choice of process and design variables for optimum process performance.

7.2 Future work and directions

Commercial PSE tools and methods that helped to develop and optimize the refinery and chemical industries were explored to simulate, analyze and optimize fermentation-based biorefineries. A novel framework of a robust optimization was linked to an unsteady state batch fermentation in Aspen Plus through a COM interface. While this general framework demonstrated its ability to serve as decision support-tool to choose, support the understanding, and offer design alternatives, there is still more room for improvement and innovation. The robustness and speed of the optimization process can be improved. For example, the design variables (elements in the population of the genetic algorithm) were passed one set at a time between MATLAB and Aspen Plus. This approach may be inefficient. Future work will focus on vectorizing the decision variables and their corresponding objective function values. In this way, the genetic algorithm will call the function computing the objective function values once to evaluate all the objectives of the individuals in the current population. This should be done in concert with the modification of the Aspen Plus file to be able to evaluate multiple decision variables at once, for example through the sensitivity analysis module.

In Chapter 4, a foundational procedure to simulate an unsteady state batch fermentation without and with *in situ* product recovery by gas stripping was developed and demonstrated to predict trends that are significantly different from the traditional separate steady state batch fermentation and product recovery. Future work will extend this simulation procedure to other fermentation schemes such as fed-batch (used to alleviate substrate inhibition) and continuous fermentation (alleviates substrate and product inhibitions). To this end, preliminary results of a fed-batch ABE fermentation and *in situ* product recovery via gas stripping, simulated with unsteady state batch reactors in series to add fresh substrate for fed-batch and add a continuous feed of gas to simulate *in situ* gas stripping are presented in Figure N.1 in Appendix N. Herein, the unsteady state fed-batch simulations using the batch ODEs alone in Aspen Plus match quantitatively with the results of the ODEs that have been explicitly modified to account for volume change due to fresh substrate addition and product separation (Figure N.2, Appendix N). This validates the procedure developed and how powerful the thermodynamic models are in traditional process simulators to predict fermentation process dynamics in terms of volume and species concentrations. The trends in the unsteady state fed-batch simulation matches the trends in fed-batch laboratory results in literature (Figure N.3, Appendix N). The fed-batch unsteady state simulation results show that the when the typical concentrations of sugars from lignocellulosic hydrolysate are used as the fed-batch stream, unrealistically large fermentor volumes will be required relative to using concentrated sugars (Figure N.4, Appendix N). Additionally, the sampling times in laboratory may be important and affect results (Figure N.5, Appendix N). These are important design revelations obtained using PSE tools and methods through *in silico* analyses for the development of fermentation-based biorefineries.

The backbone of PSE is mathematical models, which have become central to understanding cellular kinetics and mechanisms and for that matter fermentation-based biorefineries. The development of computationally tractable mathematical models should move away from steady state and stoichiometric models to mechanistic models that incorporate cellular metabolism and regulation mechanism based on extracellular changes and genetic modifications.[34] These mechanistic or kinetic models should be tailored to accommodate the typical components in using lignocellulosic biomass for biofuel productions as discussed in Chapter 3. Further, mathematical models developed for one microorganism generally lack transferability to other processes that use a different microorganism without model modification and reevaluation of the model parameters.[35] Model parameters usually change (over time), rendering the use of constant parameters in mechanistic models inadequate. An adaptive model parameter system accounting for variations in parameters based on advanced monitoring and control system can be used. It is imperative that sensitivity analyses and confidence intervals be reported with models developed.[2, 95]

In Chapter 3, the challenges for applying traditional PSE tools and methods to fermentation-based biorefineries were discussed. Through the challenges identified, traditional process simulators need to modify existing unit operations and add models that can readily simulate fermentation processes as unsteady state. This can be done by exploring avenues to extend the capabilities of unit operation models in traditional process simulators. Analytical tools that can characterize and analyze typical lignocellulosic biomass components and determine the properties of microorganisms need to be developed. In this way, the databases of commercial process simulators can be updated with typical lignocellulosic-based fermentation components. One major improvement that is urgently needed is the ability of unit operation models in these PSE tools to simulate integrated fermentation and *in situ* product recovery processes, such as adsorption, pervaporation, membrane separation, etc. which are currently either non-existent or not readily available.

Finally, the major obstacle to developing sustainable biofuels is the sheer volume of complex decisions that have to be made with respect to the use of different processing technologies, optimal resource allocation and the choice of optimal operating conditions and process inputs. The effect of uncertainty in process parameters and competing goals are exacerbated over time. For example, constant relative weights were employed in the optimization process in this work. Future work will focus on using different weights to ascertain the effect of uncertainty on the process performance. Sustainability is defined as “development that meets the need of the present without compromising the ability of future generations to meet their own needs.”[67] The three pillars of sustainability – economic, environment, and social factors must be incorporated in simultaneously making decisions about fermentation-based biorefineries. Examples of economic sustainability factors include– quality, responsiveness, and efficiency (productivity, cost reduction, service level); environment sustainability factors include – emissions, natural resource utilization, waste, and recycling; social sustainability factors include – health and safety, noise and employee well-being. As an example, the economic, environment and social factors in the form of operating cost, emissions and employee injuries were simultaneously minimized in a multi-objective optimization.[67] The challenge is circumventing the uncertainty in achieving economic viability while simultaneously meeting societal and environmental targets. The National Research Council said “Solutions to sustainability challenges typically involve finding near-optimal trade-offs among competing goals, typically under high degrees of uncertainty in both systems and the goal” and concluded that PSE tools and methods will be the gateway in addressing these challenges.[40] It is in this context that the optimization techniques developed in this study are important contributions to the development of fermentation-based biorefineries.

The use of PSE tools and methods through virtual experimentation and process analyses saves time and resources by indicating the most impactful experiments to conduct for process design, continuous improvement, and optimization. By incorporating economic, social and environment factors of sustainability in the design, analysis and optimization of fermentation-based biorefineries through commercial PSE tools and methods, the production of biofuels from lignocellulosic biomass may at last prove to be economically viable and true sustainable alternatives to fossil-based products.

Appendices

Appendix A: Ordinary differential equations representation of the fermentation kinetics of a batch culture of *Clostridium acetobutylicum*.[120]

$$\frac{dPM}{dt} = k_1 C_S \frac{K_I}{K_I + C_B} PM - 0.56(PM - 1)PM \qquad \text{(A.1)}$$

$$\frac{dC_X}{dt} = 0.56(PM - 1)C_X - k_2 C_B C_X \qquad \text{(A.2)}$$

$$\frac{dC_S}{dt} = -k_3 C_S C_X - k_4 \frac{C_S}{K_S + C_S} C_X \qquad \text{(A.3)}$$

$$\frac{dC_{BA}}{dt} = k_5 C_S \frac{K_I}{K_I + C_B} C_X - k_6 \frac{C_{BA}}{K_{BA} + C_{BA}} C_X \qquad \text{(A.4)}$$

$$\frac{dC_B}{dt} = k_7 C_S C_X - 0.841 \frac{dC_{BA}}{dt} \qquad \text{(A.5)}$$

$$\frac{dC_{AA}}{dt} = k_8 \frac{C_S}{K_S + C_S} \frac{K_I}{K_I + C_B} C_X - k_9 \frac{C_{AA}}{K_{AA} + C_{AA}} \frac{C_S}{K_S + C_S} C_X \qquad \text{(A.6)}$$

$$\frac{dC_A}{dt} = k_{10} \frac{C_S}{K_S + C_S} C_X - 0.484 \frac{dC_{AA}}{dt} \qquad \text{(A.7)}$$

$$\frac{dC_E}{dt} = k_{11} \frac{C_S}{K_S + C_S} C_X \qquad \text{(A.8)}$$

$$\frac{dC_{CO_2}}{dt} = k_{12} \frac{C_S}{K_S + C_S} C_X \qquad \text{(A.9)}$$

$$\frac{dC_{H_2}}{dt} = k_{13} \frac{C_S}{K_S + C_S} C_X + k_{14} C_S C_X \qquad \text{(A.10)}$$

Appendix B: Parameter definition for the kinetic model and their respective values.[120]

k_1	kinetic constant in Eq. A.1, = 0.009 L/g-substrate/h
k_2	kinetic constant in Eq. A.2, = 0.0008 L/g-butanol/h
k_3	kinetic constant in Eq. A.3, = 0.0255 L/g-biomass/h
k_4	kinetic constant in Eq. A.3, = 0.6764 g-substrate/g-biomass/h
k_5	kinetic constant in Eq. A.4, = 0.0136 g-butyric acid·L/g-substrate/g-biomass/h
k_6	kinetic constant in Eq. A.4, = 0.1170 g-butyric acid/g-biomass/h
k_7	kinetic constant in Eq. A.5, = 0.0113 g-butanol·L/g-substrate/g-biomass/h
k_8	kinetic constant in Eq. A.6, = 0.7150 g-acetic acid/g-biomass/h
k_9	kinetic constant in Eq. A.6, = 0.1350 g-acetic acid/g-biomass/h
k_{10}	kinetic constant in Eq. A.7, = 0.1558 g-acetone/g-biomass/h
k_{11}	kinetic constant in Eq. A.8, = 0.0258 g-ethanol/g-biomass/h
k_{12}	kinetic constant in Eq. A.9, = 0.6139 g-carbon dioxide/g-biomass/h
k_{13}	kinetic constant in Eq. A.10, = 0.0185 g-hydrogen/g-biomass/h
k_{14}	kinetic constant in Eq. A.10, = 0.00013 g-hydrogen·L /g-substrate/g-biomass/h
K_I	inhibition constant, = 0.833 g-butanol/L
K_S	Monod constant, = 2.0 g-substrate/L
K_{BA}	saturation constant, = 0.5 g-butyric acid/L
K_{AA}	saturation constant, = 0.5 L/g-acetic acid/L
C_A	acetone concentration, g/L
C_B	butanol concentration, g/L
C_E	ethanol concentration, g/L
C_{BA}	butyric acid concentration, g/L
C_{AA}	acetic acid concentration, g/L
C_S	glucose concentration, g/L
C_X	cell biomass concentration, g/L
C_{CO2}	carbon dioxide concentration, g/L
C_{H2}	hydrogen concentration, g/L
PM	marker of the physiological state culture, dimensionless

Appendix C: Description of the FORTRAN user kinetic subroutine and running the Aspen Plus RBatch block and FOTRAN code

C.1 Description of the FORTRAN user kinetic subroutine and running the Aspen Plus RBatch block

When Aspen Plus RBatch block is run, the user defined subroutine is loaded and executed dynamically. This procedure requires writing the Fortran user kinetic subroutine, compiling the written subroutine using the "aspcomp" procedure in the Aspen Plus module and supplying the object file created to the Aspen Plus system. The example, Example of a User kinetic subroutine for RBatch, in the Aspen Plus V8.8 Support Center.[122] was used as the template to write the external Fortran user kinetic subroutine. The salient parts of the template are the supplied argument list, declaration of the variables used in dimensioning the variables in the argument list, dimensioning of the variables in the argument list, accessing general labelled commons to provide physical stream property data (for further calculation in the subroutine), declaration of local variables used within the subroutine, the calculation of the reactions rates and supplying the calculated reaction rates to the argument list.

The argument list interfaces the written subroutine with the RBatch block in Aspen Plus. The variables in the argument list and variables used in dimensioning of the variables in the argument list were used as supplied in the template without any modification, as these are *fait accompli*. Stream data from the RBatch block (flow rates, parameters, and other thermodynamic properties) are accessed through general labelled commons as standard Fortran statements in the subroutine to calculate the reaction rates for each component. The reaction rates are calculated from the ordinary differential equations written in the subroutine using the stream data. The calculated reaction rates are passed through the argument list to the RBatch block to conduct a material balance to determine the condition of the fermentation broth after integration. In a cyclic communication between the RBatch and the subroutine, the time-dependent properties of the batch process are determined. The 'Customize Aspen Plus V8.8' as a module in the Aspen Plus was used to compile the written Fortran user kinetic subroutine into an object file that Aspen Plus can read using the "aspcomp" procedure.[122] Details of the FORTRAN code are in Appendix C.2.

C.2: FOTRAN code incorporating user kinetics subroutine

```
SUBROUTINE SOURCE (SOUT,   NSUBS,  IDXSUB,   ITYPE,  NINT,
     2                 INT,    NREAL,  REAL,     IDS,    NPO,
     3                 NBOPST, NIWORK, IWORK,    NWORK,  WORK,
     4                 NC,     NR,     STOIC,    RATES,  FLUXM,
     5                 FLUXS,  XCURR,  NTCAT,    RATCAT, NTSSAT,
     6                 RATSSA, KCALL,  KFAIL,    KFLASH, NCOMP,
     7                 IDX,    Y,      X,        X1,     X2,
     8                 NRALL,  RATALL, NUSERV,   USERV,  NINTR,
     9                 INTR,   NREALR, REALR,    NIWR,   IWR,
     *                NWR,    WR)
!-------------------------------------------------------------------------------
      IMPLICIT NONE

!     DECLARE VARIABLES USED IN DIMENSIONING
      INTEGER NSUBS, NINT,  NPO,   NIWORK,NWORK,
     +        NC,    NR,    NTCAT, NTSSAT,NCOMP,
     +        NRALL, NUSERV,NINTR, NREALR,NIWR,
     +        NWR

!- Arrays in Argument List. The stream vector's dimension will
!be determined by Aspen Plus at run-time.
!------------------------------------------------------------------------
!- Labeled commons are provided to pass data for specific reactor types.

!- RBATCH
#include "rbtc_rbati.cmn"
#include "rbtc_rbatr.cmn"

!- Pressure Relief
#include "rbtc_presrr.cmn"

!- for console
#include "dms_maxwrt.cmn"

!- for components
#include "dms_ncomp.cmn"

! Retrieving phase properties
#include "flsh_prplus.cmn"

      EQUIVALENCE (AMWL, PRPLUS_AMWL)
!------------------------------------------------------------------------
!- Two general labeled commons are provided for physical property data.
#include "rxn_rprops.cmn"
      EQUIVALENCE (TEMP, RPROPS_UTEMP)    !TEMP = Reactor temperature,
      EQUIVALENCE (PRES, RPROPS_UPRES)    !PRES = Reactor Pressure
      EQUIVALENCE (VFRAC, RPROPS_UVFRAC)  !VFRAC = Reactor Molar vapor fraction
      EQUIVALENCE (BETA, RPROPS_UBETA)    !BETA = Reactor total liquid molar ratio
      EQUIVALENCE (VVAP, RPROPS_UVVAP)    !VVAP = Reactor vapor phase volume(m3)
      EQUIVALENCE (VLIQ, RPROPS_UVLIQ)    !VLIQ =Reactor liquid phase volume(m3)
      EQUIVALENCE (VLIQS, RPROPS_UVLIQS)  !VLIQS = volume occupied by the liquid
and solid phases in the reactor(m3)
      EQUIVALENCE (VOLRB, RBATR_VOLRB)    !VOLRB=Reactor liquid phase volume(m3)
#include "pputl_ppglob.cmn"
```

```
!------------------------------------------------------------------------
!- A general user common is provided for all user-specified routines.
! RMISS = Real missing value, IMISS = integer missing value
#include "ppexec_user.cmn"
      EQUIVALENCE (RMISS, USER_RUMISS)
      EQUIVALENCE (IMISS, USER_IUMISS)
!------------------------------------------------------------------------
!- Commons and Declarations for the Aspen Plus Error Reporting Facility.
#include "dms_plex.cmn"
      REAL*8 B(1)
      EQUIVALENCE (B,IB)
#include "dms_errout.cmn"
      EQUIVALENCE (IERROUT, ERROUT_IEROUT)

!     DECLARE ARGUMENTS
      INTEGER IDXSUB(NSUBS),ITYPE(NSUBS), INT(NINT),
     +        IDS(2),NBOPST(6,NPO),IWORK(NIWORK),
     +        IDX(NCOMP),   INTR(NINTR),  IWR(NIWR),
     +        NREAL, KCALL, KFAIL, KFLASH, I, LMW,
     +        IBIO, IS, IBA, IBU, IA, IAA, IETOH, IH2O, IPBIO
      INTEGER IPS, IPBA, IPB, IPA, IPAA, IPETOH, IPH2O, KV, KDIAG,
     +        KER, IPYM, IPCO2, IPH2, IYM, ICO2, IH2, IN2,IPN2
      REAL*8 SOUT(1),      WORK(NWORK),
     +       STOIC(NC,NSUBS,NR),  RATES(NC),
     +       FLUXM(1),     FLUXS(1),     RATCAT(NTCAT),
     +       RATSSA(NTSSAT),      Y(NCOMP),
     +       X(NCOMP),     X1(NCOMP),    X2(NCOMP)
      REAL*8 RATALL(NRALL),USERV(NUSERV),
     +       REALR(NREALR),WR(NWR), XCURR, TEMP, AMWL, AMW,
     +       PRES,  VMXL,  DVMX,  TK , CQ (NCOMP),  MWAL

!     DECLARE LOCAL VARIABLES
      INTEGER IPROG(2),IMISS, DMS_KFORMC,DMS_IRRCHK, DMS_IFCMNC
      REAL*8 REAL(NREAL),   VFRAC, BETA, VOLRB,
     + VVAP,  VLIQ,  VLIQS, RMISS, CBIO, CS, CBA, CB, CX, A,
     +    CA, CAA, CETOH, CH2O,  RRATE1, RRATE2, RRATE3, RRATE4,RRATE5,
     +    RRATE6, RRATE7, RRATE8, RRATE9, RRATE10, CYM, CCO2, CH2,CN2,
     +    CXO,YM,DY, CCYM,CCH2,CCCO2, CCS,CCB, CCBA, CCA, CCAA,CCETOH,
     +    CSMW, CBAMW, CBMW,CAMW,CAAMW,CETOHMW, XTIME,CBIOMW,CH2OMW,
     +    CCH2O, CCBIO, CYMMW, CH2MW,CCO2MW,U, LIQDEN, SUM,CN2MW, CCN2

      CHARACTER*80 IERROUT(10), IERW1(10), IERW2(9), IERW3(8)
     +             , IERW4(7), IERW5(6), IERW6(5), IERW7(4), IERW8(3)
     +             , IERW9(2), IERW10

      EQUIVALENCE (IERROUT(1), IERW1), (IERROUT(2), IERW2),
     +            (IERROUT(3), IERW3), (IERROUT(4), IERW4),
     +            (IERROUT(5), IERW5), (IERROUT(6), IERW6),
     +            (IERROUT(7), IERW7), (IERROUT(8), IERW8),
     +            (IERROUT(9), IERW9), (IERROUT(10), IERW10)


!- Declaration for Aspen Plus diagnositic reporting routine.
!     DATA STATEMENTS
      INTEGER MODELS(5)
      DATA MODELS /4HRBAT,4HRCST,4HRPLU,4HPRES,4HRADF/
      DATA IPROG /4HUSRK, 4HIN  /
!===============================================================================
```

```fortran
!- Initialize rates vector.
      DO I = 1, 2*NC
        RATES(I) = 0.
      END DO
!==============================================================================
! Locate component indices used in this routine based on formula name.
!    It enables the user routine to automatically use the correct indices
!    if the component order is changed, or if components are inserted or
!    deleted from the components paragraph.
!==============================================================================
      IBIO =DMS_KFORMC ('CHX')
      IS =DMS_KFORMC ('C6H12O6')
      IBA  =DMS_KFORMC ('C4H8O2-1')
      IBU   =DMS_KFORMC ('C4H10O-1')
      IA =DMS_KFORMC ('C3H6O-1')
      IAA =DMS_KFORMC ('C2H4O2-1')
      IETOH  =DMS_KFORMC ('C2H6O-2')
      IH2O   =DMS_KFORMC ('H2O')
      IYM   =DMS_KFORMC ('MAKER')
      IH2   =DMS_KFORMC ('H2')
      ICO2   =DMS_KFORMC ('CO2')
      IN2 = DMS_KFORMC ('N2')
!==============================================================================
! Locate the packed component indices.
!==============================================================================
       DO I=1, NCOMP
           IF (IDX(I).EQ.IBIO) THEN
                IPBIO=I
           ELSE IF (IDX(I).EQ.IS) THEN
                IPS=I
           ELSE IF (IDX(I).EQ.IBA) THEN
                IPBA=I
           ELSE IF (IDX(I).EQ.IBU) THEN
                IPB=I
           ELSE IF (IDX(I).EQ.IA) THEN
                IPA=I
           ELSE IF (IDX(I).EQ.IAA) THEN
                IPAA=I
           ELSE IF (IDX(I).EQ.IETOH) THEN
                IPETOH=I
           ELSE IF (IDX(I).EQ.IH2O) THEN
                IPH2O=I
          ELSE IF (IDX(I).EQ.IYM) THEN
                IPYM=I
           ELSE IF (IDX(I).EQ.IH2) THEN
                IPH2=I
           ELSE IF (IDX(I).EQ.ICO2) THEN
                IPCO2=I
           ELSE IF (IDX(I).EQ.IN2) THEN
                IPN2=I
           END IF
       END DO
!==============================================================================

! Calculate the concentration of each component.
!==============================================================================
              KV=1
       !Mixture molar volume (Liquid)
       CALL PPMON_VOLL ( TEMP  , PRES , X , NCOMP , IDX, NBOPST,
     +       KDIAG, KV, VMXL, DVMX, KER )

      !kgmol /m3.h
      CCBIO = SOUT(IBIO)/VLIQS! Units of VLIQS= m3
      CCS = SOUT(IS)/VLIQS
      CCBA = SOUT(IBA)/VLIQS
      CCB = SOUT(IBU)/VLIQS
      CCA = SOUT(IA)/VLIQS
      CCAA = SOUT(IAA)/VLIQS
      CCETOH = SOUT(IETOH)/VLIQS
      CCH2O = SOUT(IH2O)/VLIQS
      CCYM = SOUT(IYM)/VLIQS
      CCH2 = SOUT(IH2)/VLIQS
      CCCO2 = SOUT(ICO2)/VLIQS
      CCN2 = SOUT(IN2)/VLIQS

      U = SOUT(NCOMP_NCC+8) ! MASS DENSITY OF THE OUTLET STREAM IN KG/M3
!===============================================================================
!     Convert concentration to g/s.L
!     Get Molecular weights of components
       LMW = DMS_IFCMNC('MW')
       CBIOMW  = B(LMW + IBIO);
       CSMW = B(LMW + IS);
       CBAMW = B (LMW + IBA);
       CBMW  = B(LMW + IBU);
       CAMW  = B(LMW + IA);
       CAAMW = B(LMW + IAA);
       CETOHMW = B (LMW + IETOH);
       CH2OMW  = B(LMW + IH2O);
       CYMMW = B(LMW + IYM);
       CH2MW = B (LMW + IH2);
       CCO2MW  = B(LMW + ICO2);
      CN2MW = B(LMW + IN2);

      !(CONCENTRATION)*(METER T0 L CONVERSION)* (MW)* (g to kg conversion)
      ! g/L.h = (kg-mol/h/M3)*(1M3/1000L)*(kg/kg-MOLE)* (1000g/kg)
      CS = CCS*CSMW
      CBA = CCBA*CBAMW
      CA = CCA*CAMW
      CAA = CCAA*CAAMW
      CB = CCB*CBMW
      CETOH = CCETOH*CETOHMW
      CH2O = CCH2O*CH2OMW
      CBIO = CCBIO*CBIOMW
      CYM = CCYM*CYMMW
      CH2 = CCH2*CH2MW
      CCO2 = CCCO2*CCO2MW
      CN2=CCN2*CN2MW

      !Convert time to hours
      XTIME = XCURR/3600

      ! Model equations from (J. VOTRUBA, B. VOLESKY AND L. YERUSHALMI,
MATHEMATICAL MODEL
      !OF A batch acetone-butanol fermentation. Biotechnology AND BIOENGINEERING,
1986. 23(9): p. 247-255.)
      ! Calculate Rates in g/L.h
      ! Here, REALR(i) present the kinetic parameters in the ODEs, values were
entered in Aspen Plus and called here.
      RRATE1 = (REALR(1)*CS*REALR(15)*CYM/(REALR(15)+CB))
     +        -(0.56*CYM*(CYM-1))                             ! PHYSIOLOGICAL
MAKER
      RRATE2 = (0.56*CBIO*(CYM-1))-(REALR(2)*CB*CBIO)          ! CELL BIOMASS
      RRATE3 = (-REALR(3)*CS*CBIO)-(REALR(4)*CBIO*CS/
     +         ((CS+REALR(16))))                               ! GLUCOSE
      RRATE4 = (REALR(5)*CS*CBIO*REALR(15)/(REALR(15)+CB))-
     +         (REALR(6)*CBIO*CBA/((CBA+REALR(17))))           !BUTYRIC ACID
      RRATE6 = ((REALR(8)*CS*REALR(15)*CBIO)/((CS+REALR(16))
     +        *(CB+REALR(15))))-((REALR(9)*CAA*CS*CBIO)/((CAA+REALR(18))
     +         *(CS+REALR(16))))                                          !ACETIC
ACID

      RRATE5 = (REALR(7)*CS*CBIO)-(0.841*RRATE4)                          !BUTANOL
      RRATE7 = (REALR(10)*CS*CBIO/(REALR(16)+CS))-(0.484*RRATE6)          !ACETONE
      RRATE8 = (REALR(11)*CS*CBIO/(REALR(16)+CS))
      RRATE9 = (REALR(12)*CS*CBIO/(REALR(16)+CS))                         !CARBON
DIOXIDE
      RRATE10 = (REALR(13)*CS*CBIO/(REALR(16)+CS))+(REALR(14)*CS*CBIO)
!HYDROGEN

!===============================================================================
 ! Recovert rates from g/(L.h) to kmol/s : (g/L/h)*/(kg/kmol)*(1kg/1000g)*(1000
L/m3)*(1h/3600sec)
      RATES(IYM) = RRATE1*VLIQS/(CYMMW*3600)
      RATES(IBIO) = RRATE2*VLIQS/(CBIOMW*3600)
      RATES(IS) = RRATE3*VLIQS/(CSMW*3600)
      RATES(IBA) = RRATE4*VLIQS/(CBAMW*3600)
      RATES(IBU) = RRATE5*VLIQS/(CBMW*3600)
      RATES(IAA) = RRATE6*VLIQS/(CAAMW*3600)
      RATES(IETOH) = RRATE8*VLIQS/(CETOHMW*3600)
      RATES(IA) = RRATE7*VLIQS/(CAMW*3600)
      RATES(ICO2) = RRATE9*VLIQS/(CCO2MW*3600)
      RATES(IH2) = RRATE10*VLIQS/(CH2MW*3600)
!==============================================================================
    ! Printing to the user variables table
      USERV(1) = U                 ! MIXTURE DENSITY, KG/M3
      USERV(2) = VOLRB*1000 !TOTAL REACTOR VOLUME, LITERS
      USERV(3) =VLIQ*1000          !LIQUID PHASE VOLUME, LITERS
      USERV(4) = VLIQS*1000 !LIQUID AND SOLID PHASE, LITERS
      USERV(5) = VVAP*1000  !VAPOR PHASE VOLUME, LITERS
      USERV(6) = LIQDEN            !DENSITY OF THE LIQUID, g/L or kg/m3
      USERV(7) = PRES/101325       !PRESSURE IN atm
      USERV(8) =X(IPBIO)           !LIQUID MOLE FRACTION OF CELL BIOMASS
      RETURN
      END
!===============================================================================
```

Appendix D: Procedure for the compilation of the Fortran user kinetic subroutine code to a readable Aspen Plus format

Aspen Plus cannot read raw Fortran codes; the code must be transformed to a format (.object file) that Aspen Plus can relate and run. Aspen Plus ‘Customize Aspen Plus V8.8’ as module in the Aspen package was used to compile the Fortan code into an object file that Aspen plus can read. The Fortran code and aspen (.apw or .bkp) file must be located in the same folder. For example, the commands to compile the Fortran subroutine for a folder located on a desktop using ‘Customize Aspen Plus V8.8’ window is:

```
cd c://users/username/desktop/ Folder name (hit enter)
```

If the folder location appears in the next line, Aspen Plus was able to locate the folder with the next line as: c://users/username/desktop/Folder name

The command ‘aspcomp Filename’ was typed as: c://users/username/desktop/Folder name/ aspcomp FORTRAN FILE NAME (hit enter).

If there is no error in the file, the file will be compiled and .object file will appear in the same folder.

Appendix E: Stoichiometric equations and coefficients used in simulating a steady state batch fermentation in Aspen Plus, using a stoichiometric reactor

Stoichiometric equations (equations E.1 – E.5) used together with stoichiometric coefficients relative to glucose.[5, 87, 118-119] The stoichiometric coefficients used in the stoichiometric reactor were 0.319, 0.495, 0.080, 0.120, 0 (mole of product/mole of glucose fed) for acetone, butanol, ethanol, acetic and butyric acids, respectively, calculated from the model of Votruba et al.[120]

$C_6H_{12}O_6 \rightarrow C_4H_{10}O$ (butanol) $+ 2CO_2 + H_2O$ (E.1)

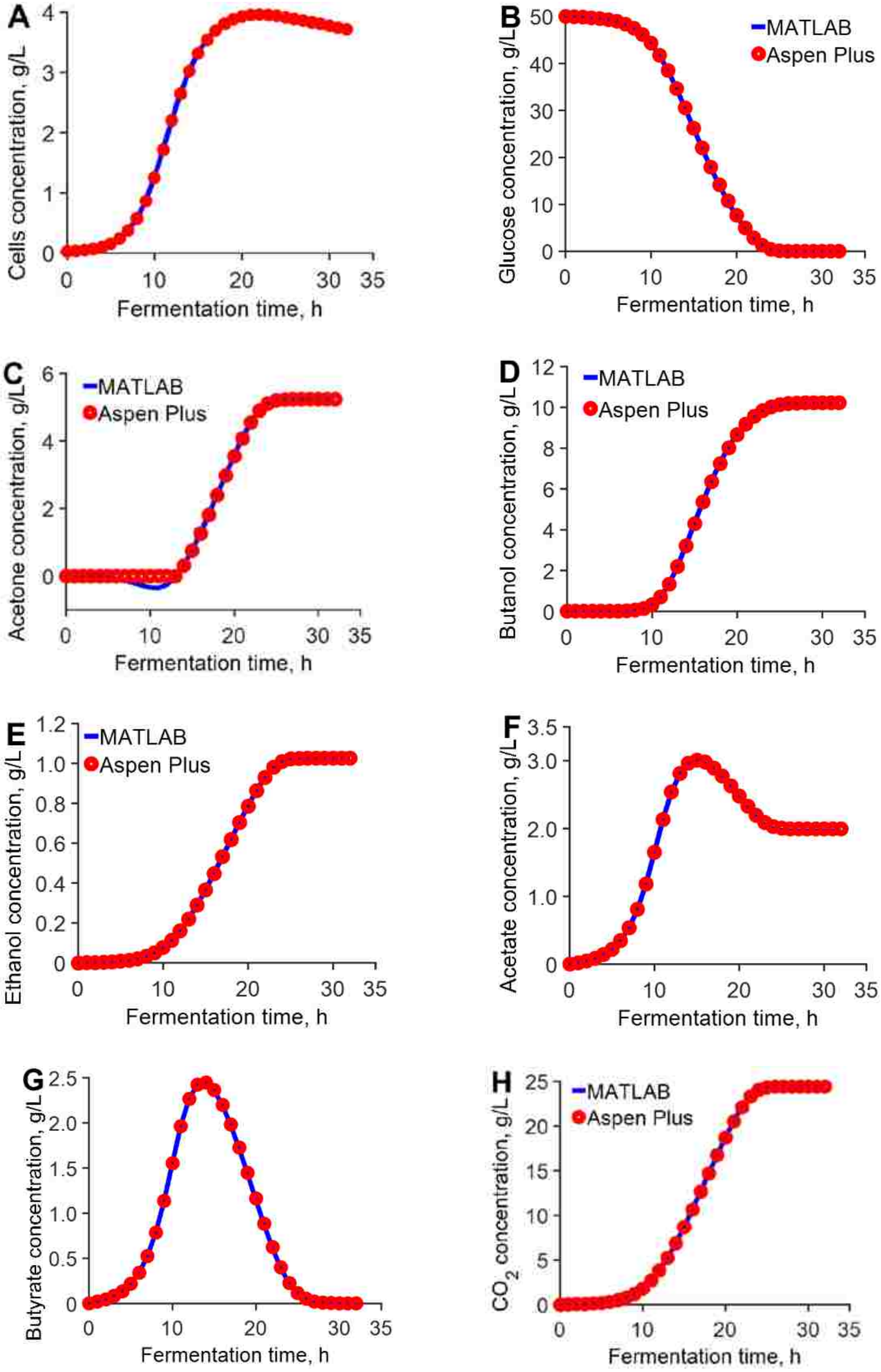
$C_6H_{12}O_6 + H_2O \rightarrow C_3H_6O$ (acetone) $+ 3CO_2 + 4H_2$ (E.2)

$C_6H_{12}O_6 \rightarrow 2C_2H_5O$ (ethanol) $+ 2CO_2 + H_2$ (E.3)

$C_6H_{12}O_6 \rightarrow C_4H_8O_2$ (butyric acid) $+ 2CO_2 + 2H_2$ (E.4)

$C_6H_{12}O_6 \rightarrow 3C_2H_4O_2$ (acetic acid) (E.5)

Appendix F: Graphs comparing simulation trends in Aspen Plus RBatch with the integration of ODEs describing the batch fermentation process in MATLAB.



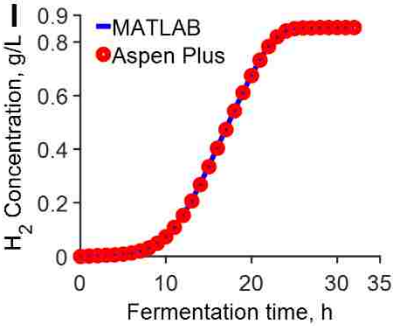


Figure F.1 Comparison of the simulation results of the RBatch block in Aspen Plus with the integration of ODEs describing the batch fermentation process in MATLAB
Cells (A), Glucose (B), Acetone(C), Butanol (D), Ethanol (E), Acetic Acid (F), Butyric Acid (G), Carbon Dioxide (H) and Hydrogen (I)

Appendix G: Comparison of simulation trends using the traditional Aspen Plus steady state approach that simulate the fermentation process entirely separate from the gas stripping process with the integrated batch ABE and *in situ* gas stripping simulated with the RBatch block linked to the Fortran user kinetic subroutine.

The batch fermentation was simulated with a stoichiometric reactor, RStoic, using product yields (mole of product/mole of glucose fed) of 0.319, 0.495, 0.080, 0.120, 0 for acetone, butanol, ethanol, acetic and butyric acids, respectively. The final composition from the RStoic model was fed to a flash unit using different gas flow rates (0.8, 1.6, 3, 5, 6.4 L/min per L of broth).

Table G.1 Selectivity, percent recovery and condensate concentration of the separate steady state fermentation and gas stripping.

Gas flow ate	Selectivity				Recovery,%				Condensate concentration, g/L			
L/min per L broth	Acetone	Butanol	Ethanol	ABE	Acetone	Butanol	Ethanol	ABE	Acetone	Butanol	Ethanol	ABE
0.8	44.4	4.4	10.0	19.1	6.2	0.7	1.5	2.6	193.8	41.3	18.6	253.7
1.6	44.3	4.4	10.0	19.1	6.4	0.7	1.6	2.7	193.5	41.4	18.6	253.4
3.0	44.1	4.5	9.9	19.0	6.8	0.7	1.6	2.9	193.0	41.4	18.6	253.0
5.0	43.9	4.5	9.9	19.0	7.3	0.8	1.8	3.1	192.3	41.4	18.6	252.3
6.4	43.8	4.6	9.9	18.9	7.6	0.8	1.9	3.2	191.8	41.5	18.6	251.9

Table G.2 Selectivity, percent recovery and condensate concentration of the integrated batch ABE and *in situ* gas stripping simulated with the RBatch block linked to the Fortran user kinetic subroutine (one RBatch block for 32 h fermentation time) using different gas flow rates (0.8, 1.6, 3, 5, 6.4 L/min per L of broth). Initial reactor conditions: 50 g/L glucose and 0.03 g/L cell biomass.

Gas flow rate	Selectivity				Recovery, %				Condensate concentration, g/L			
L/min per L broth	Acetone	Butanol	Ethanol	ABE	Acetone	Butanol	Ethanol	ABE	Acetone	Butanol	Ethanol	ABE
0.8	77.4	8.1	9.8	18.0	79.9	30.1	35.4	46.4	79.1	57.2	6.8	143.0
1.6	218.9	10.1	13.0	19.7	96.1	53.0	60.1	67.3	50.2	52.8	6.0	109.1
3	1855.1	14.8	20.9	25.2	99.8	78.1	83.9	85.4	28.9	44.0	4.8	77.6
5	76570.4	28.3	48.2	44.5	100.0	93.6	96.3	95.7	17.5	33.4	3.5	54.4
6.4	1783313.0	49.4	99.2	76.0	100.0	94.2	97.0	96.1	13.6	27.9	2.8	44.3

Appendix H: Comparison of the selectivity and percent recovery of acetone, butanol, ethanol, acetic acid, butyric acid and water at different gas flow rates per L of the fermentation broth for a batch fermentation and *in situ* gas stripping process.

Table H.1 Selectivity and percent recovery of the Aspen Plus RBatch ABE fermentation and *in situ* gas stripping simulation with 4 RBatch blocks connected in series. Gas flow rates used: 1, 3, 5 L/min per L of fermentation broth and gas was fed at 15, 20, 25 h.

Selectivity						
Time, h	Acetone	Butanol	Ethanol	Acetate	Butyrate	Water
1L/min N_2 per L of broth						
10	0.0	3.6	6.7	0.3	0.4	18.3
15	16.4	4.6	6.9	0.4	0.6	0.9
20	32.6	6.5	7.7	0.4	0.9	0.1
25	50.7	7.8	9.4	0.3	3.0	0.1
32	76.9	9.1	11.4	0.3	76.8	0.1
3 L/min N_2 per L of broth						
10	0.0	3.6	6.7	0.3	0.4	18.3
15	16.4	4.6	6.9	0.4	0.6	0.9
20	41.3	7.3	8.7	0.4	0.9	0.2
25	90.3	9.6	11.9	0.4	4.5	0.1
32	407.4[a]	12.4	16.9	0.4	370.3[a]	0.1
5L/min N_2 per L of broth						
10	0.0	3.6	6.7	0.3	0.4	18.3
15	16.4	4.6	6.9	0.4	0.6	0.9
20	47.6	8.0	9.6	0.4	0.9	0.2
25	175.3	11.4	14.7	0.4	6.9	0.1
32	4505.0[a]	17.2	26.3	0.4	2586.7[a]	0.2

Percent Recovery						
Time, h	Acetone	Butanol	Ethanol	Acetate	Butyrate	Water
1L/min N_2 per L of broth						
10	0.0	0.0	0.1	0.0	0.0	0.0
15	0.6	0.2	0.3	0.0	0.0	0.0
20	35.9	10.4	12.4	0.7	1.6	1.6
25	47.2	12.7	15.5	0.7	5.5	1.6
32	65.5	18.5	22.9	0.9	66.9	2.3
3 L/min N_2 per L of broth						
10	0.0	0.0	0.1	0.0	0.0	0.0
15	0.6	0.2	0.3	0.0	0.0	0.0
20	69.1	28.5	33.0	2.2	5.0	4.9
25	84.7	36.9	43.0	2.3	22.3	5.5
32	97.5	54.1	62.3	3.5	97.3	8.6
5 L/min N_2 per L of broth						
10	0.0	0.0	0.1	0.0	0.0	0.0
15	0.6	0.2	0.3	0.0	0.0	0.0
20	80.8	41.3	46.6	3.4	7.9	7.8
25	94.4	52.1	59.3	3.5	40.6	8.5
32	99.9	71.9	79.9	5.4	99.7	12.9

[a] Component almost exhausted in the fermenter.

Appendix I: Concentration of ABE, acetic and butyric acids in the condensate (stripped stream) and reactor using different gas flow rates per L of the fermentation broth for a batch fermentation and *in situ* gas stripping process.

Similar to batch fermentation and *in situ* gas stripping experiments[42], 4 RBatch blocks were connected in series in order to collect the stripped stream in 4 different sets. Gas flow rates used: 1, 3, 5 L/min per L of fermentation broth and gas was fed after 15 h.

Table I.1 Comparison of the condensate and reactor concentrations ABE fermentation and *in situ* gas stripping simulation.

Concentration, g/L										
Time, h	Acetone		Butanol		Ethanol		Acetic acid		Butyric acid	
	Reactor	Stripped	Reactor	Stripped	Reactor	Stripped	Reactor	Stripped	Reactor	Stripped
1L/min N_2 per L of broth										
10	0.0	0.0	0.3	1.2	0.1	0.5	1.7	0.4	1.6	0.6
15	1.1	17.2	4.3	19.1	0.4	2.5	3.0	1.0	2.4	1.4
20	2.5	74.2	7.8	48.1	0.7	5.2	2.5	1.0	1.2	1.0
25	2.2	98.7	8.1	60.0	0.8	7.3	2.1	0.7	0.1	0.3
32	0.8	56.8	6.6	59.8	0.6	7.1	2.1	0.7	0.0	0.0
3 L/min N_2 per L of broth										
10	0.0	0.0	0.3	1.2	0.1	0.5	1.7	0.4	1.6	0.6
15	1.1	17.2	4.3	19.1	0.4	2.5	3.0	1.0	2.4	1.4
20	1.2	49.9	6.3	45.6	0.5	4.8	2.6	1.0	1.1	1.1
25	0.5	43.5	5.1	51.2	0.5	5.9	2.2	0.9	0.1	0.3
32	0.0	6.7	2.6	37.6	0.2	3.9	2.2	1.0	0.0	0.0
5 L/min N_2 per L of broth										
10	0.0	0.0	0.3	1.2	0.1	0.5	1.7	0.4	1.6	0.6
15	1.1	17.2	4.3	19.1	0.4	2.5	3.0	1.0	2.4	1.4
20	0.7	36.4	5.1	42.5	0.4	4.4	2.6	1.1	1.1	1.1
25	0.1	24.1	3.1	41.1	0.3	4.6	2.4	1.1	0.0	0.3
32	0.0	0.9	0.9	21.0	0.1	1.9	2.3	1.2	0.0	0.0

Appendix J: Formulation of the MOO and SOO problems used in case study in Chapter 5.

$MOO: Maximize$

$Total\ ABE\ produced, g/L = \sum_{f=1}^{g}\left[\frac{(C_{f,re}V_{re}+C_{f,st}V_{st})}{(V_{re}+V_{st})}\right]$

$ABE\ concentration\ in\ condensate, g/L = \sum_{f=1}^{g} C_{f,st}$

$ABE\ Selectivity = y(1-x)/x(1-y)$

$where\ y = \frac{C_{f,st}V_{st}}{\sum_{h=1}^{g} C_{h,st}V_{st}}\ and\ x = \frac{C_{f,re}V_{re}}{\sum_{h=1}^{g} C_{h,re}V_{re}}$ (J.1)

$Subject\ to$:

$Gas\ flow\ rate\ (F)\ 0.1 \leq F \leq 5\ L\,\mathrm{min}\,per\ L\ of\ broth$

$Gas\ stripping\ initiation\ time\ (FT)\ 15 \leq FT \leq 25\ h$

$SOO: Maximize$

$\sum_{l=1}^{g} w_l Z_l$ (J.2)

$Subject\ to$:

$Gas\ flow\ rate\ (F)\ 0.1 \leq F \leq 5\ L\,\mathrm{min}\,per\ L\ of\ broth$

$Gas\ stripping\ initiation\ time\ (FT)\ 15 \leq FT \leq 25\ h$

where f = Acetone, butanol and ethanol, h = acetone, butanol, ethanol, acetic acid, butyric acid and water, l = ABE produced, ABE concentration in condensate and ABE Selectivity, re = reactor, st = stripped stream (condensate), g = number of components, w = relative weight of each objective function, C = concentration (g/L), V = volume in liters, Z = Objective function value.

Appendix K: Details of the Net Flow Method (NFM)[138] and MATLAB sample code for scoring a set of Pareto optimal points using the NFM.

K.1: Details of the Net Flow Method (NFM).[138]

A Pareto domain is a set of alternative solutions obtained in the simultaneous optimization of multiple objectives. Multi-objective optimization (MOO) can optimize multiple and competing objectives, generates solutions that span the entire decision (search) space, find multiple global or local optima (global solution) and does not require functional derivatives. After generating a large number of non-dominated solutions in MOO process, the solutions in the Pareto domain are scored and ranked using preferences and relative weights based on the expertise of the decision-maker. For example, a genetic algorithm can be used to generate a large number of non-dominated solutions. The Net Flow Method (NFM) and Rough Set Method (RSM) are two examples used to rank and score the solutions in the Pareto domain.

NFM uses the expertise of the decision maker to score and tank the solutions in the Pareto domain that are equally optimal. NFM uses three thresholds and a set of relative weights for each objective function to score and rank the solutions in the Pareto domain. The indifference threshold represents a value below which if the difference between two objective function values falls, the two objective function values cannot be distinguished. This means, one objective function value cannot be chosen over another if the difference of the objective function values is below the indifference threshold. If the difference between a pair of objective function values is greater than the preference threshold, the better solution is chosen. For example, for a maximization problem, the preferred value will be the objective function with the larger value and vice versa. When the difference between two objective function values is greater than the veto threshold, one solution is banned against the other solution. The indifference, preference and veto thresholds are established such that 0 ≤ indifference threshold ≤ preference threshold ≤ veto threshold.

NFM algorithm.

1. ***Relative difference,*** $\Delta_k[i,j]$. The relative difference between the values F_k of each objective function k is calculated when solution i is compared with solution j (within the same objective function):

$$\Delta_k[i,j] = F_k(i) - F_k(j) \begin{cases} i \in [1, M] \\ j \in [1, M] \\ k \in [1, M] \end{cases}$$

where M is the solution in the Pareto domain. For maximization, $-\Delta_k[i,j]$ are used in the subsequent equations whereas $\Delta_k[i,j]$ is used for minimization. When the objective is to meet a target value, $F_k(i)$ and $F_k(j)$ correspond to the absolute difference between solution k and the corresponding target values, in which case $\Delta_k[i,j]$ is used.

2. ***Individual concordance index,*** $c_k[i,j]$**.** The individual concordance indices are determined for all n objectives and for each pair of solution using the relative difference ($\Delta_k[i,j]$), the indifference (Q_k) and preference thresholds (P_k):

$$c_k[i,j] = \left\{ \begin{array}{cc} 1 & \text{if } \Delta_k[i,j] \le Q_k \\ \dfrac{P_k - \Delta_k[i,j]}{P_k - Q_k} & \text{if } Q_k \le \Delta_k[i,j] \le P_k \\ 0 & \text{if } \Delta_k[i,j] > P_k \end{array} \right\}$$

The relationship between the individual concordance index ($c_k[i,j]$), the calculated relative differences ($\Delta_k[i,j]$), the indifference (Q_k) and preference (P_k) thresholds is illustrated in Figure K.A

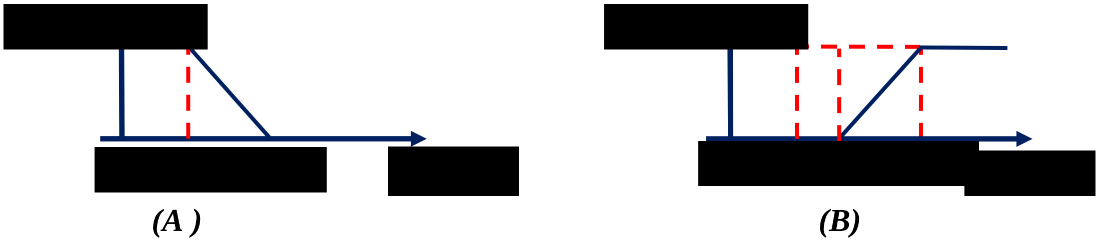


Figure K.1 Illustration of the (A) concordance index, and (B) discordance index calculations used in the NFM algorithm to score and rank the solutions in the Pareto domain.

3. ***Global concordance index,*** $C_k[i,j]$**.** The global concordance indices are calculated as the weighted sum of the individual concordance indices as follows, using the relative weights (W_k) of the objective function values:

$$C_k[i,j] = \sum_{k=1}^{n} W_k c_k[i,j] \quad \left\{ \begin{array}{l} i \in [1, M] \\ j \in [1, M] \end{array} \right\}$$

4. ***Discordance index,*** $D_k\left[i, j\right]$**.** The discordance index is calculated for each objective function, *k*, using the relative difference ($\Delta_k\left[i, j\right]$), together with the preference (*P*$_k$) and veto (*V*$_k$) thresholds:

$$D_k\left[i, j\right] = \left\{\begin{matrix} 0 & if\ \ \Delta_k\left[i, j\right] \le P_k \\ \dfrac{\Delta_k\left[i, j\right] - P_k}{V_k - P_k} & if\ \ \mathrm{P}_k \le \Delta_k\left[i, j\right] \le V_k \\ 1 & if\ \ \Delta_k\left[i, j\right] > V_k \end{matrix}\right\}$$

The relationship between the individual discordance index ($D_k\left[i, j\right]$), the calculated relative differences ($\Delta_k\left[i, j\right]$), the preference (*Q*$_k$) and veto (*V*$_k$) thresholds is illustrated in Figure K.B.

5. ***Outranking matrix,*** $\sigma\left[i, j\right]$. The outranking matrix is calculated for each pair of solution to determine the relative performance of a pair of solutions in the Pareto domain using the global concordance ($C_k\left[i, j\right]$) and discordance ($D_k\left[i, j\right]$) indices:

$$\sigma\left[i, j\right] = C_k\left[i, j\right]\left(\prod_{k=1}^{n}\left[1 - \left(D_k\left[i, j\right]\right)^3\right]\right) \quad \left\{\begin{matrix} i \in \left[1, M\right] \\ j \in \left[1, M\right] \end{matrix}\right\}$$

The outranking matrix measures the performance of solution *i* relative to solution *j* (for all *n* objectives) in the Pareto domain. If $\sigma\left[i, j\right]$ is close to 0, solution *j* outranks solution *i* whereas $\sigma\left[i, j\right]$ close to 1 indicates solution *i* outranks solution *j*.

6. ***Final score,*** σ_i. A final score for each solution in the Pareto domain is determined by the sum of the individual outranking matrix values, (by taking the difference in the sum of all the elements in the column by the sum of all elements in the respective rows).

$$\sigma_i = \sum_{j=1}^{M}\sigma\left[i, j\right] - \sum_{j=1}^{M}\sigma\left[j, i\right]$$

The first term determines how solution *i* performs relative to all other solutions in the Pareto domain while the second term determines the performance of all other solutions relative to solution *i*.

7. ***Ranking solutions in the Pareto domain***

   The calculated scores are then arranged from the highest to the lowest. The best optimal solution is the solution with the highest score based on the ranking criteria set by the decision maker (relative weights and preference thresholds).

K.2: MATLAB sample code for scoring a set of Pareto optimal points using the NFM.

```
format short

% Objectives being maximized (insert objectives in this format)
maxobj = [0.4791    1.1289  71.3975 0.6187  ;
0.478   2.3795  107.5435    1   ;
0.478   2.7402  217.931 1   ;
0.4787  1.1392  74.4217 0.6259  ;
0.478   2.0354  156.009 1   ;

               ⋮

0.5227  0.3673  30.7483 0.1003  ;
0.5264  0.2144  17.8524 0.057   ;
0.4985  1.5452  63.9676 0.2422  ;
0.478   2.1948  160.3775    1   ];

% Relative weights (Wk) for each objective (Relative importance of each
% objective function or criteria, Sum(Wi) = 1)
wk = [1/4 1/4 1/4 1/4];

% Indifference threshold, qk,(The range of variation of each criterion for which
% it impossible for the decision maker to choose the criterion of one
% solution over the corresponding criterion of another solution. In other
% words the range of values over which two objectives function values are
% indiscernable). Taken as 5% of the range of each objective function values.
qk =[0.0027 0.1426  11.0676 0.0495];

% Preference threshold, pk, (Preference is given to the solution with a
% better criterion if the difference between two values for a given
% criterion exceeds the preference threshold. For maximization, the better
% solution is the larger values and vice versa for minimization).
%Taken as 10% of the range of each objective function values.
pk = [0.0054    0.2853  22.1351 0.0990];

% Veto threshold, vk, (This threshold eliminates on solution relative to
% another if the difference between the values of their respective
% objective function is too high. The solution is eliminated if at least
% one of the objective function values voilates the veto threshold even if
% other objective funciton values are accepted.
%Taken as 30% of the range of each objective function values.

vk = [0.0162    0.8558  66.4054 0.2969];

% 0<= qk<=pk<=vk

m = size(maxobj,1); % Number of rows
n = length(maxobj); % Number of columns
y = length(wk);
% Preallocate for speed.
deltak1=zeros(n,n);
deltak2=zeros(n,n);
 deltak3=zeros(n,n);
 deltak4=zeros(n,n);
for i = 1:n
    for j = 1:n
        deltak1(i,j) =  bsxfun(@minus,maxobj(j,1),maxobj(i,1));
        deltak2(i,j) = bsxfun(@minus,maxobj(j,2),maxobj(i,2));
         deltak3(i,j) =  bsxfun(@minus,maxobj(j,3),maxobj(i,3));
         deltak4(i,j) = bsxfun(@minus,maxobj(j,4),maxobj(i,4));
    end

end

% For each objective function, solution i is compared to solution j and the
% difference calculated.
deltakij1= deltak1;
deltakij2= deltak2;
 deltakij3= deltak3;
 deltakij4= deltak4;

% For each objective function, the individual concordance index ck[i,j] for
% each criterion is determined for all n objective criteria and for each
% pair of solutions
ck1=zeros(n,n);
for i = 1:m
    for j = 1:m
if deltakij1(i,j)<= qk(1,1)
    ck1(j,i) = 1;
elseif deltakij1(i,j) > pk(1,1)
    ck1(j,i) = 0;
 else
    ck1(j,i) = (pk(1,1)-deltakij1(i,j))./( pk(1,1)-qk(1,1));
end
    end
end

ck2=zeros(n,n);
for i = 1:m
    for j = 1:m
if deltakij2(i,j)<= qk(1,2)
    ck2(j,i) = 1;
elseif deltakij2(i,j) > pk(1,2)
    ck2(j,i) = 0;
 else
    ck2(j,i) = (pk(1,2)-deltakij2(i,j))./( pk(1,2)-qk(1,2));
end
    end
end

ck3=zeros(n,n);
for i = 1:m
    for j = 1:m
if deltakij3(i,j)<= qk(1,3)
    ck3(j,i) = 1;
elseif deltakij3(i,j) > pk(1,3)
    ck3(j,i) = 0;
 else
    ck3(j,i) = (pk(1,3)-deltakij3(i,j))./( pk(1,3)-qk(1,3));
end
    end
end

ck4=zeros(n,n);
for i = 1:m
    for j = 1:m
if deltakij4(i,j)<= qk(1,4)
    ck4(j,i) = 1;
elseif deltakij4(i,j) > pk(1,4)
    ck4(j,i) = 0;
 else  %qk(1,1) < deltakij1 <= pk(1,1)
    ck4(j,i) = (pk(1,4)-deltakij4(i,j))./( pk(1,4)-qk(1,4));
end
    end
end

% Individual concordance index
ckij1 = ck1;
ckij2 = ck2;
 ckij3 = ck3;
 ckij4 = ck4;
```

```
%Global concordance index (C(i,j)) calculated when comparing solution i and
%j where the weighted sum of the inidividual concordance indices is calculated.
C =zeros(n,n);
for i = 1:m
    for j = 1:m
        C(i,j) =
ckij1(i,j)*wk(1,1)+ckij2(i,j)*wk(1,2)+ckij3(i,j)*wk(1,3)+ckij4(i,j)*wk(1,4);
%
    end
end
cij = C;

dk1=zeros(n,n);
for i = 1:m
    for j = 1:m
if deltakij1(i,j)<= pk(1,1)
    dk1(j,i) = 0;
elseif deltakij1(i,j) > vk(1,1)
    dk1(j,i) = 1;
 else
    dk1(j,i) = (deltakij1(i,j)-pk(1,1))./( vk(1,1)-pk(1,1));
end
    end
end

dk2=zeros(n,n);
for i = 1:m
    for j = 1:m
if deltakij2(i,j)<= pk(1,2)
    dk2(j,i) = 0;
elseif deltakij2(i,j) > vk(1,2)
    dk2(j,i) = 1;
 else
    dk2(j,i) = (deltakij2(i,j)-pk(1,2))./( vk(1,2)-pk(1,2));
end
    end
end

dk3=zeros(n,n);
for i = 1:m
    for j = 1:m
if deltakij3(i,j)<= pk(1,3)
    dk3(j,i) = 0;
elseif deltakij3(i,j) > vk(1,3)
    dk3(j,i) = 1;
 else
    dk3(j,i) = (deltakij3(i,j)-pk(1,3))./( vk(1,3)-pk(1,3));
end
    end
end
%
dk4=zeros(n,n);
for i = 1:m
    for j = 1:m
if deltakij4(i,j)<= pk(1,4)
    dk4(j,i) = 0;
elseif deltakij4(i,j) > vk(1,4)
    dk4(j,i) = 1;
 else
    dk4(j,i) = (deltakij4(i,j)-pk(1,4))./( vk(1,4)-pk(1,4));
end
    end
end

% Individual disconcordance index, Dik
dkij1 = dk1;
dkij2 = dk2;
dkij3 = dk3;
 dkij4 = dk4;
```

```
% Outranking matrix S, Sigma(i, j), measures the quality of solution i
% relative to solution j in terms of n objective functions. An element of
% Sigma(i,j) close to ) means solution j outranks solution i. However, if
% the value of the outranking matrix is close to 1, solution i may outrank
% solution j or may be located in the neighborhood of solution j.

% Preallocate for speed
S =zeros(n,n);
for i = 1:m
    for j = 1:m
       S(i,j) = cij(i,j).*((1-((dkij1(i,j)^3)))*(1-((dkij2(i,j)^3)))*((1-
((dkij3(i,j)^3))))*(1-((dkij4(i,j)^3))));
    end
end
sij = S;

% The final ranking score (Si) for each solution in the Pareto domain, Sigma i,
% is obtained by summing the individual outranking elements associated with
% each domain solution.

Si=zeros(length(S),1);
z = 1:m;
for i = 1:m
      Si(i) = sum(sij(z,i))-sum(sij(i,z));
end
 Rank = Si
```

Appendix L: ODEs representation of the batch ethanolic fermentation kinetics of a batch culture of *Z. mobilis* and parameter definition for the kinetic model and their respective values.[137]

$$\frac{dm_{Q,G}}{dt} = \mu_{max,G}\left(\frac{m_G}{K_{SQ,G}+m_G}\right)\left(1-\frac{m_{ET}-m_{TIQ,G}}{m_{MQ,G}-m_{TIQ,G}}\right)\left(\frac{K_{IQ,G}}{K_{IQ,G}+m_G}\right) \tag{L.1}$$

$$\frac{dm_{Q,H}}{dt} = \mu_{max,H}\left(\frac{m_H}{K_{SQ,H}+m_H}\right)\left(1-\frac{m_{ET}-m_{TIQ,H}}{m_{MQ,H}-m_{TIQ,H}}\right)\left(\frac{K_{IQ,H}}{K_{IQ,H}+m_H}\right) \tag{L.2}$$

$$\frac{dm_Q}{dt} = \left[\eta\frac{dm_{Q,G}}{dt} + (1-\eta)\frac{dm_{Q,H}}{dt}\right].m_Q \tag{L.3}$$

$$\frac{dm_G}{dt} = -\eta\varphi_{Smax,G}\left(\frac{m_G}{K_{SS,G}+m_G}\right)\left(1-\frac{m_{ET}-m_{TIS,G}}{m_{MS,G}-m_{TIS,G}}\right)\left(\frac{K_{IS,G}}{K_{IS,G}+m_G}\right) \tag{L.4}$$

$$\frac{dm_H}{dt} = -(1-\eta)\varphi_{Smax,H}\left(\frac{m_H}{K_{SS,H}+m_H}\right)\left(1-\frac{m_{ET}-m_{TIS,H}}{m_{MS,H}-m_{TIS,H}}\right)\left(\frac{K_{IS,H}}{K_{IS,H}+m_H}\right) \tag{L.5}$$

$$\frac{dm_{ET,G}}{dt} = \varphi_{ETmax,G}\left(\frac{m_G}{K_{SET,G}+m_G}\right)\left(1-\frac{m_{ET}-m_{TIET,G}}{m_{MET,G}-m_{TIET,G}}\right)\left(\frac{K_{IET,G}}{K_{IET,G}+m_G}\right) \tag{L.6}$$

$$\frac{dm_{ET,H}}{dt} = \varphi_{ETmax,H}\left(\frac{m_H}{K_{SET,H}+m_H}\right)\left(1-\frac{m_{ET}-m_{TIET,H}}{m_{MET,H}-m_{TIET,H}}\right)\left(\frac{K_{IET,H}}{K_{IET,H}+m_H}\right) \tag{L.7}$$

$$\frac{dm_{ET}}{dt} = \left[\eta\frac{dm_{ET,G}}{dt} + (1-\eta)\frac{dm_{ET,H}}{dt}\right].m_Q \tag{L.8}$$

Where:

$\frac{dm_{Q,G}}{dt}$ = Cell biomass growth rate based on glucose, g/(L.h)

$\frac{dm_{Q,H}}{dt}$ = Cell biomass growth rate based on xylose, g/(L.h)

$\frac{dm_Q}{dt}$ = Total cell growth rate based on both glucose and xylose, g/(L.h)

$\frac{dm_G}{dt}$ = Glucose consumption rate, g/(L.h)

$\frac{dm_H}{dt}$ = Xylose consumption rare, g/(L.h)

$\frac{dm_{ET,G}}{dt}$ = Ethanol production based on glucose, g/(L.h)

$\frac{dm_{ET,H}}{dt}$ = Ethanol production based on xylose, g/(L.h)

$\frac{dm_{ET}}{dt}$ = Total ethanol production rate based on both glucose and xylose, g/(L.h)

m_Q = Cell biomass concentration, g/L

m_G = Glucose concentration, g/L

m_H = Xylose concentration, g/L

m_{ET} = Ethanol concentration, g/L

$\mu_{max,G}$ = maximum specific cell growth based on glucose, 0.31 h^{-1}

$K_{SQ,G}$ = Monod saturation constant/substrate limitation constant for cell growth based on glucose, 1.45 g/L

$m_{TIQ,G}$ = Threshold inhibitory ethanol concentration for cell growth based on glucose, 28.9 g/L

$m_{MQ,G}$ = Maximum inhibitory ethanol concentration for cell growth based on glucose, 57.2 g/L

$K_{IQ,G}$ = Substrate inhibition constant in cell growth based on glucose, 200 g/L

$\mu_{max,H}$ = maximum specific cell growth based on xylose, 0.1 h^{-1}

$K_{SQ,H}$ = Monod saturation constant/substrate limitation constant for cell growth based on xylose, 4.91 g/L

$m_{TIQ,H}$ = Threshold inhibitory ethanol concentration for cell growth based on xylose, 26.6 g/L

$m_{MQ,H}$ = Maximum inhibitory ethanol concentration for cell growth based on xylose, 56.3 g/L

$K_{IQ,H}$ = Substrate inhibition constant in cell growth based on xylose, 600 g/L

$\varphi_{Smax,G}$ = Maximum specific glucose utilization, 10.9 g/(g.h)

$K_{SS,G}$ = Substrate limitation constant for glucose utilization, 6.32 g/L

$m_{TIS,G}$ = Threshold inhibitory ethanol concentration for glucose consumption, 42.6 g/L

$m_{MS,G}$ = Maximum inhibitory ethanol concentration for glucose consumption, 75.4 g/L

$K_{IS,G}$ = Substrate inhibition constant in glucose consumption, 186 g/L

$\varphi_{Smax,H}$ = Maximum specific xylose utilization, 3.27 g/(g.h)

$K_{SS,H}$ = Substrate limitation constant for xylose utilization, 0.03 g/L

$m_{TIS,H}$ = Threshold inhibitory ethanol concentration for xylose consumption, 53.1 g/L

$m_{MS,H}$ = Maximum inhibitory ethanol concentration for xylose consumption, 81.2 g/L

$K_{IS,H}$ = Substrate inhibition constant in xylose consumption, 600 g/L

$\Phi_{ETmax,G}$ = Maximum specific ethanol production based on glucose fermentation, 5.12 g/(g.h)

$K_{SET,G}$ = Substrate limitation constant for glucose fermentation, 6.32 g/L

$m_{TIET,G}$ = Threshold inhibitory ethanol concentration for glucose fermentation, 42.6 g/L

$m_{MET,G}$ = Maximum inhibitory ethanol concentration for glucose fermentation, 75.4 g/L

$K_{IET,\ G}$ = Substrate inhibition constant in glucose fermentation, 186 g/L

$\Phi_{ETmax,\ H}$ = Maximum specific ethanol production based on xylose fermentation, 1.59 g/(g.h)

$K_{SET,\ H}$ = Substrate limitation constant for xylose fermentation, 0.03 g/L

$m_{TIET,\ H}$ = Threshold inhibitory ethanol concentration for xylose fermentation, 53.1 g/L

$m_{MET,\ H}$ = Maximum inhibitory ethanol concentration for xylose fermentation, 81.2 g/L

$K_{IET,\ H}$ = Substrate inhibition constant in xylose fermentation, 600 g/L

η = Weighting factor for glucose consumption, 0.65

Appendix M: Creating a two-way communication between Aspen Plus and MATLAB to write and read data between the two platforms

The example “Aspen Plus COM automation using MATLAB” with article ID 000026382 in Aspen Plus support center was used as a guide. The syntax to create the COM interface, open Aspen Plus, write data from MATLAB to Aspen Plus to, read data from Aspen Plus to MATLAB, reinitialize, run, save and close Aspen Plus are described in the following sections.

M.1 Creation of automation server with a handle to the COM interface

Aspen = actxserver (‘apwn.document.35.0’)

where 35.0 is the internal version number of Aspen Plus V9.

M.2 Opening Aspen Plus

A batch reactor linked to the Fortran user kinetics subroutine was initially created and saved. The path to the backup simulation file (AspenPath) is supplied to the invoking syntax to open the Aspen Plus simulation file every time it is needed.

Invoke (Aspen, 'InitFromFile2', AspenPath)

M.3. Writing and reading data between Aspen Plus and MATLAB

The COM interface establishes a two-way communication between Aspen Plus and MALAB. To write data from MATLAB to Aspen Plus and read data from Aspen Plus to MATLAB, the path to the Node is required. To establish this path in Aspen Plus V9 for example, the data of interest is found through Customize/Variable Explorer/Data (navigating to the data of interest).

To change the value of a variable located in the path Node “writevariablepath” to ***alpha***, the following syntax is used

Aspen.Application.Tree.FindNode (‘writevariablepath’).Value = ***alpha***

To read the value of a variable from Aspen Plus and assign to a variable, ***Beta***, in MATLAB with a path to node “readvariablepath”

Beta = Aspen.Application.Tree.FindNode (‘readvariablepath’).Value

M.4 Syntaxes for Reinitializing, Running, Saving and Closing Aspen Plus

Aspen.Reinit ()
Aspen.Run2 ()
Aspen.Save ()
Aspen.Quit (); delete (Aspen)

Appendix N: Preliminary Aspen Plus results of an unsteady state fed-batch without and with in situ separations (gas stripping).

The batch reactor in Aspen Plus was linked to a FORTRAN user kinetics subroutine (using techniques developed in Chapter 4) to incorporate the ordinary differential equations representing the fermentation kinetics of a batch culture of *Clostridium acetobutylicum*[120] presented in Appendices A and B. These unsteady state batch reactor representations were connected in series, where for example RBATCH1 in Figure N.1 represents the initial batch process of 50 g/L glucose and 0.03 g/L cells (run until glucose is almost used up, for about 20 h). Similar to laboratory fed-batch experiments,[10] a concentrated glucose solution of 500 g/L was fed in a short time (about 0.001 h) to bring the concentration of glucose to about 50 g/L (RBATCH2 in Figure N.1). To simulate an unsteady state fed-batch and *in situ* gas stripping process, a continuous feed of N_2 gas was fed to the batch reactor (RBATCH3) connected in series to RBATCH2. Subsequently, units of 2 unsteady state batch reactors (representing fresh substrate addition and integrated fed-batch and in situ gas stripping) were used to simulate an unsteady state fed-batch fermentation and in situ gas stripping process.

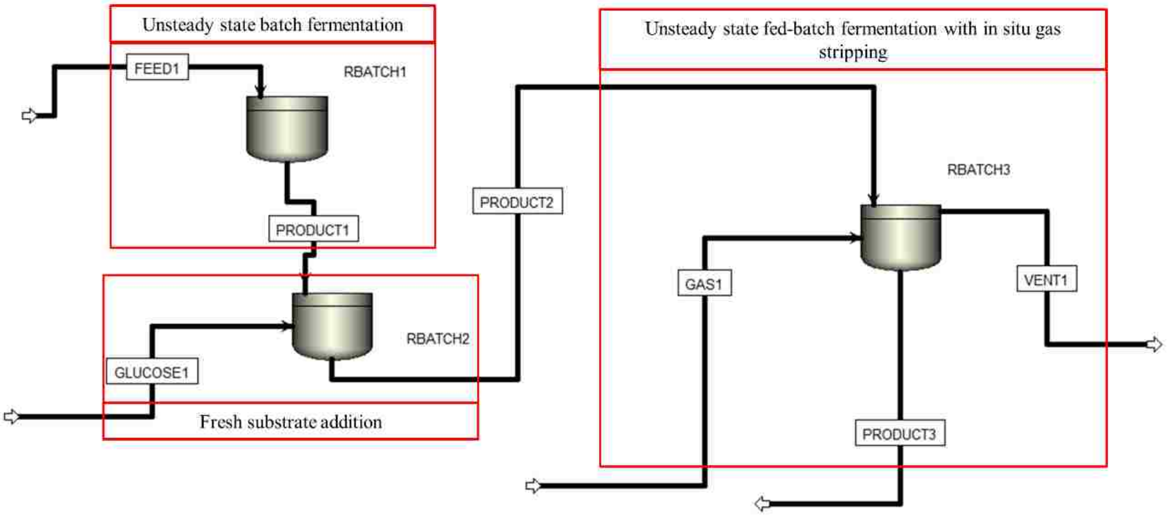

Unsteady state batch fermentation
FEED1
RBATCH1
PRODUCT1
RBATCH2
GLUCOSE1
Fresh substrate addition
PRODUCT2
Unsteady state fed-batch fermentation with in situ gas stripping
RBATCH3
GAS1
VENT1
PRODUCT3

Figure N.1 Simulation of fed-batch and *in situ* gas stripping using unsteady state batch reactors in series with a gas continuously fed.

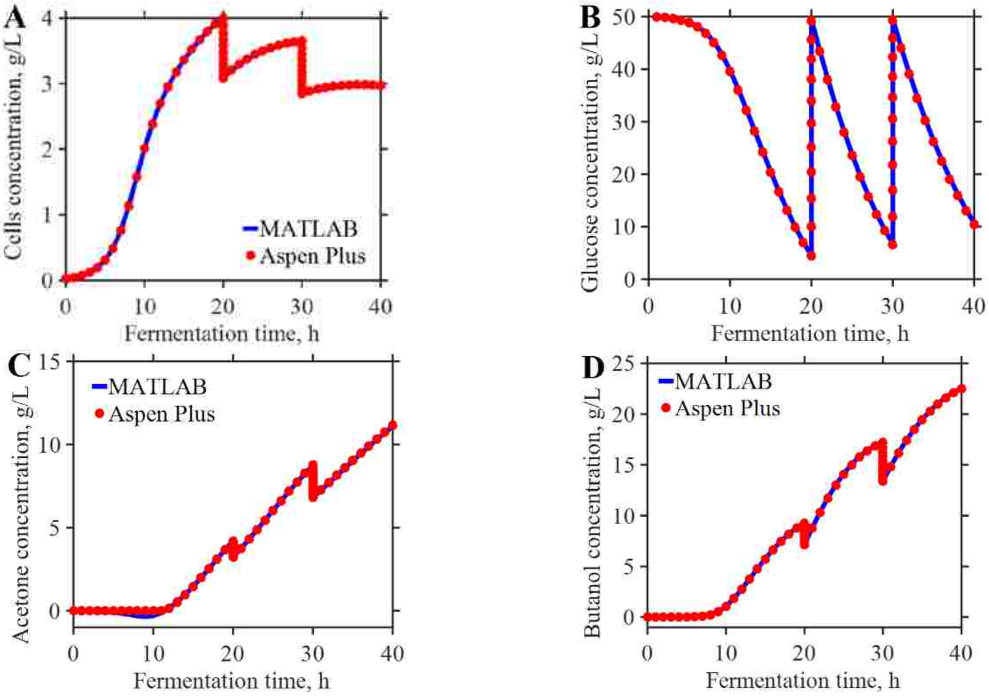


Figure N.2 Selected results showing that Aspen Plus and MATLAB fed-batch results match, validating the procedure using 500 g/L concentrated glucose solution as fed-batch stream. MATLAB ODEs were modified to account for volume change. Aspen Plus used batch model without explicitly accounting for volume change

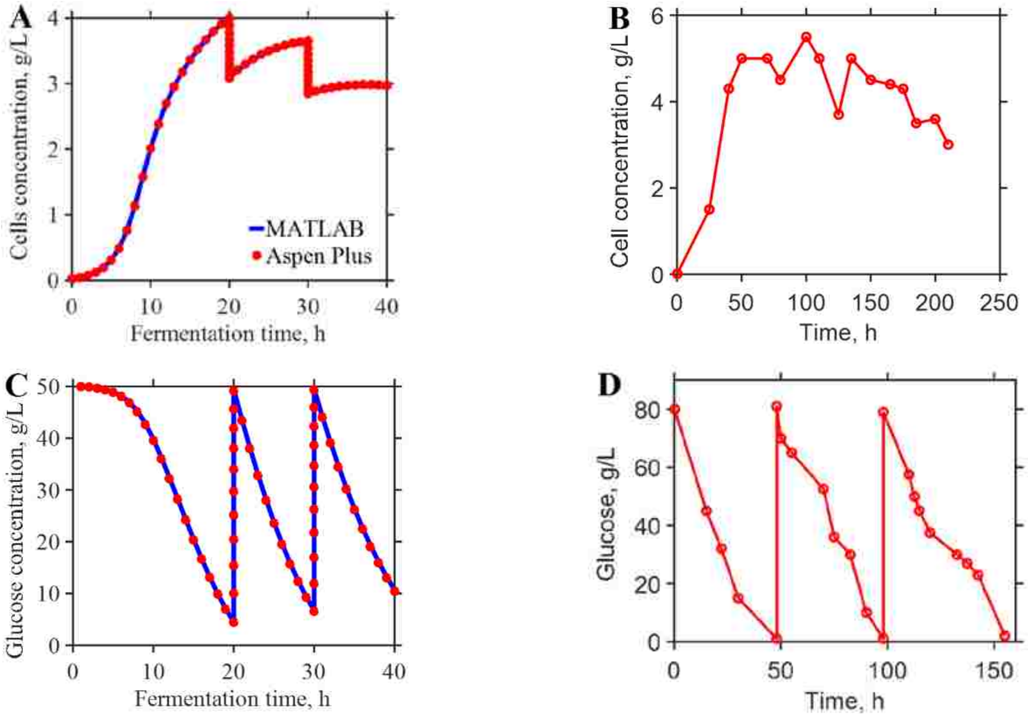


Figure N.3 Unsteady state fed-batch and *in situ* gas stripping trends match trends in integrated batch and in situ gas stripping results in literature (B[1] and D[141]).

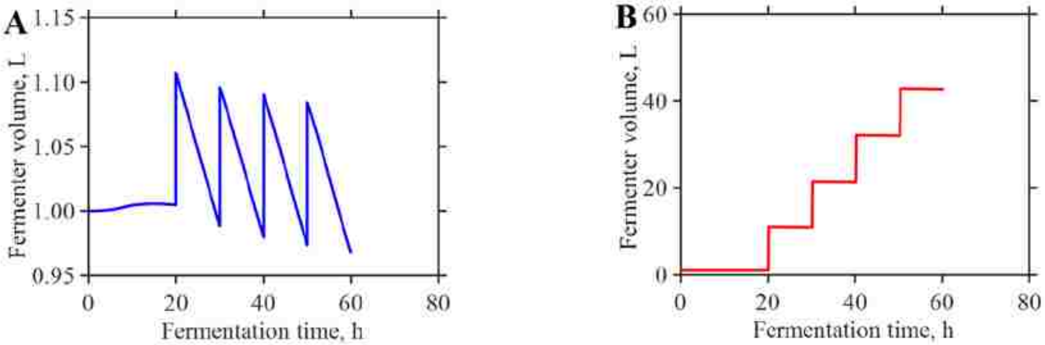


Figure N.4 Comparison of the changes in the liquid volume in the reactor for (A) using 500 g/L concentrated glucose solution and (B) 50 g/L in an unsteady fed-batch simulation in Aspen Plus to bring the concentration of glucose to about 50 g/L.

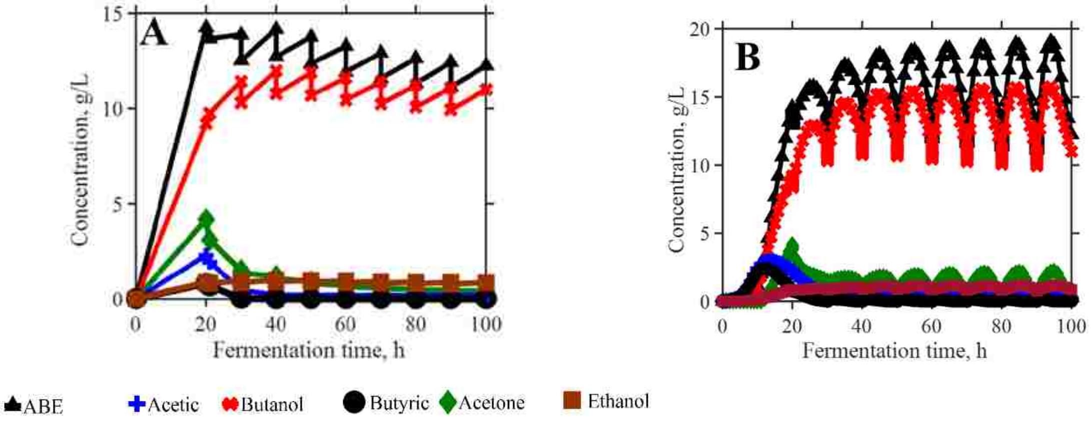


Figure N.5 Trends in the unsteady state simulation results for a fed-batch with *in situ* gas stripping (started after 20 h of fermentation) using 3L/min per L of broth. 500 g/L concentrated glucose solution was used to readjust the substrate concentration when it was almost used up. (A) Data simulated every 10 hrs. (B) Data simulated every hour.

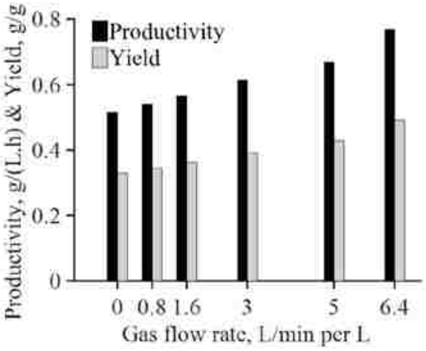


Figure N.6 The effect of gas flow rate on the ABE productivity and yield for unsteady state simulation fed-batch with *in situ* gas stripping (started after 20 h of fermentation). 500 g/L concentrated glucose solution was used to readjust the substrate concentration when it was almost used up.

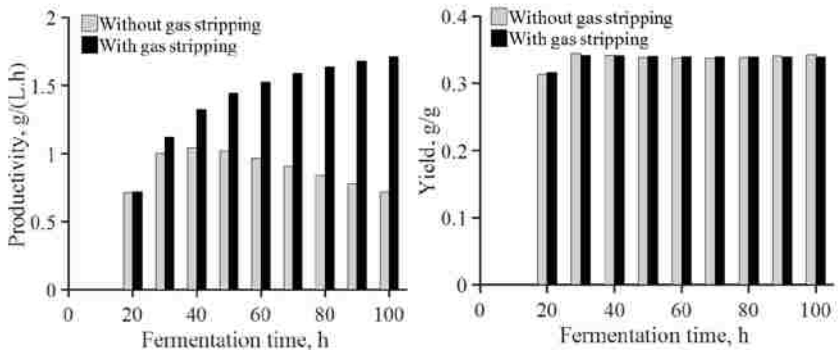


Figure N.7 Comparison of the productivity and yield of the fed-batch without and with in situ gas stripping. Gas stripping started after 20 h of fermentation using 3L/min per L of broth. 500 g/L concentrated glucose solution was used to readjust the substrate concentration when it was almost used up.

Appendix O: MOO and SOO code for the batch ABE fermentation with *in situ* gas stripping used in the Chapter 5 for the case study

O.1: Main MOO code

```
function main
clc;clear; close all
set(0,'defaultAxesFontSize', 32)
set(0,'defaultlinelinewidth',3)

opts =
optimoptions(@gamultiobj,'PlotFcn',{@gaplotpareto},'FunctionTolerance',0.01,'MaxGeneratio
ns',30,'display', 'iter',...
    'OutputFcns', @outputfunction,'PopulationSize',500,'StallGenLimit',5)

rng default % for reproducibility % Random number generator
%rng('default') puts the settings of the random number generator used by
    %RAND, RANDI, and RANDN to their default values so that they produce the
    %same random numbers as if you restarted MATLAB. In this release, the
    %default settings are the Mersenne Twister with seed 0.
tic
% Call the function  'Calculation' that calculates the optimal values
[Decison_Variables,Objective_Function,eflag,outpt,population, score] = Calculation(opts)
toc

% Get a list of the final set of decision variables
x1=Decison_Variables(:,1)./0.06583 % Gas flow rate in kg/hr converted to L/min
x2=Decison_Variables(:,2)% Gas stripping start time, h

% Get the objective values
Objective_1 = -Objective_Function(:,1) % ABE concentration in condensate, g/L
Objective_2 = -Objective_Function(:,2) % ABE Selectivity
Objective_3 = -Objective_Function(:,3)  % Total ABE produced, g/L

End


% Function Calculation takes the input opts to give the decision variables
% (x), objective function values (f) etc.
function [x,f,eflag,outpt,pop,score] = Calculation(opts)

%Fun calls the function objfun to evaluate the objective function values
%for each pair of decision values generated within the range of the
%decision values specified
fun = @objfun; % the objective function, nested below

%lower and upper bounds of the decsion variables
lb = [0.006583 15]; % Gas stripping time started after 15 to 25 hrs of fermentation
ub = [0.32915 25]; % Gas flowrate of N2 0.1 to 5 L/min per L of broth converted
% x1 is gas flow rate and x2 is gas stripping start time

% gamultiobj uses the function fun and the in-built solver to determine the
% Pareto front and the optimal values.
tic
[x,f,eflag,outpt,pop,score] = gamultiobj(fun,2,[],[],[],[],lb,ub,[],opts)
toc

 function CCC =objfun(x)
 attempt = 1;  % Preallocate the first attempt
 trying = true; %Set while to true
 % The try catch loop runs the code in the try block until an error
 % occurs,in which case the code passes the excecution to the catch block.
 % The catch block gives information on what type of error there is through
 % ME and gives instructions on how to handle the error until it is
 % rectified. This prevents the code from stopping and loosing data
 while trying
    try
     BBB=@objfunction; % Set a handle to call the objective function to run normally
     CCC=BBB(x); % The objective function values are evaluated at the decision values (x)
     trying = false; % If there is an error and trying is set to false.

    catch ME
    disp(ME) %Displays the error
    fprintf('Failed attempt (%d). Retyring...',attempt)
    pause(1)
     attempt = attempt+1; % Sets attempt to 2 and objfunj to rerun the decision value
                          % pair of points that run into error.
    end

 end
    end


    function AAA = objfunction(x)
    format long
% Print out the current pair of decision values being evaluated for the objective
fucntion.
input =x

% Calculation of the objective functions

AspenVersion = 'apwn.document.35.0'; % Programme ID (Progid) of Aspen Plus document
class.
% Also 'apwn.document.34' to specify V8.8 as the version to be used.
%V9 is internal version 35, so you should use 35 instead of 34 for the document class

AspenPath = 'C:\Users\kda228\Desktop\RBatchtoMATLAB\votruba.bkp'; % Replace with the
simulation path where file  Aspen Plus file is located
AspenVisible = 1; % 0 for not visible (Make Aspen Plus file visible when MATLAB code is
run)
AspenDialogs = 1; % 0 for not visible (Make Aspen Plus Diagnositic page visible when
MATLAB code is run)

% Get pointer and load Aspen Plus comserver:
% actxserver = creates and manipulates objects from Matlab thar are exposed
% in an application that supports automation
Aspen = actxserver(AspenVersion);
% Aspen = a handle to the default interface of the server (actxserver)

% get = Used to Get property value from interface, or display properties
get(Aspen)

% Open the Aspen-Simulation
invoke(Aspen,'InitFromArchive2',AspenPath);

Aspen.visible = AspenVisible; % Make it visible
Aspen.SuppressDialogs = AspenDialogs; % Make simulation run messages visible

% Read the Feed temperature and pressure from Aspen Plus heater input
% Gets the whole set of properties for the specified node including
% (Application, Parent, Name, Dimension, ValueType, Value, Elements, UnitSting)

%ReadFeedTempFromAspen=Aspen.Application.Tree.FindNode('\Data\Streams\FEED\Input\TEMP\MIX
ED').get

%ReadFeedTempFromAspen2=Aspen.Application.Tree.FindNode('\Data\Streams\FEED\Input\TEMP\MI
XED').Value
  %Aspen.Application.Tree.FindNode('\Data\Streams\FEED\Input\TEMP\MIXED').Value
  % Note that Aspen = A handle to the default actxserver server interface
  %Once navigation is made to the variable of interest, the path next to
  %"call" in the Variable Explorer is added to the default interface handle
  % Note the example,
  % Application.Tree.FindNode("\Data\Streams\FEED\Input\TEMP\MIXED") double
  % quotation marks "\Data\Streams\..." should be changed to single
  % quotation marks '\Data\Streams\FEED\Input\TEMP\MIXED' in MATLAB every
  % time the 'call path' from Aspen Plus is to be used in MATLAB

   % Set new values of the decision value
  NewGasflowmassrate= x(1); % Gas flow rate in kg/h
  NewStartTime = (x(2));   %Gas stripping start time h
  % Use the new gas stripping start time to set up the times for Aspen Plus
  Newasfeedtime1=(NewStartTime-0.001); % Substract 0.001 from the time as the first time
slot, h (Trick used to make sure gas stripping times are right)
  Newasfeedtime2=NewStartTime; % Actual gas stripping start time, h
  Gasflowrate = (NewGasflowmassrate*1000/(60*1.097190232)); % Gas flow rate in L/min

  % Set the gas flow rate on the continous feed at the second gas stripping
  % start time to the new mass flow rate of the gas in kg/h
  Aspen.Application.Tree.FindNode('\Data\Blocks\RBATCH1\Input\FPROF_FLOW\GAS\#1').Value =
NewGasflowmassrate;

  % Read the old gas stripping start time from the previous run
  oldtime1=
Aspen.Application.Tree.FindNode('\Data\Blocks\RBATCH1\Input\FPROF_TIME\GAS\#0').Value;

  %Compare the old old stripping start time from the previous run and the
  %new gas stripping start time to determine whether to write first to the first
  %time or second slot. This avoid an error where Aspen Plus does not allow
  %a smaller number to be written to the second time slot when the previous
  %run had a value bigger than the current time being written in the first time slot.
  if NewStartTime >oldtime1
  Aspen.Application.Tree.FindNode('\Data\Blocks\RBATCH1\Input\FPROF_TIME\GAS\#1').Value =
Newasfeedtime2; % Write 2nd time slot first
  Aspen.Application.Tree.FindNode('\Data\Blocks\RBATCH1\Input\FPROF_TIME\GAS\#0').Value =
Newasfeedtime1; %Write 1st time slot second
  else
  Aspen.Application.Tree.FindNode('\Data\Blocks\RBATCH1\Input\FPROF_TIME\GAS\#0').Value =
Newasfeedtime1; % write 1st time slot first
  Aspen.Application.Tree.FindNode('\Data\Blocks\RBATCH1\Input\FPROF_TIME\GAS\#1').Value =
Newasfeedtime2; %Write 2nd time slot second
  end

  % Set the new gas flow rate to the gas flow rate from the current pair of
  % point in L/min

Aspen.Application.Tree.FindNode('\Data\Streams\GAS\Input\TOTFLOW\MIXED').Value=Gasflowrat
e; % New Feed mass fraction of Ethanol

 %Reinit and run simulation
 Aspen.Reinit();  %

%Run the simulation case, asynchronously if the argument is True. If
%the user interface is visible, simulations should always be run
%asynchronously. Asynchronously allows the automation client to proceed with
%other tasks while waiting for the simulationrun to complete.
%Aspen Plus should always be run asynchronously if the application is visible.
 Aspen.Run2();

% Save the results of the current run in Aspen Plus
Aspen.Save();

% Read the new data (output) results from Aspen Plus
  VLIQS=
Aspen.Application.Tree.FindNode('\Data\Blocks\RBATCH1\Output\USER\#3\32\VLIQS\LITERS').Va
lue;% Volume of the liquid in the reactor, Liters

 Factor=1000/60; %Convert mass from kg/hr to g/min;

AinReactor=(Aspen.Application.Tree.FindNode('\Data\Streams\REALIQ\Output\MASSFLOW\MIXED\A
').Value)*Factor; % Mass of acetone in reactor,g/min

BinReactor=(Aspen.Application.Tree.FindNode('\Data\Streams\REALIQ\Output\MASSFLOW\MIXED\B
').Value)*Factor; % Mass of butanol in reactor,g/min

EinReactor=(Aspen.Application.Tree.FindNode('\Data\Streams\REALIQ\Output\MASSFLOW\MIXED\E
TOH').Value)*Factor; % Mass of ethanol in reactor,g/min


AinVent=(Aspen.Application.Tree.FindNode('\Data\Streams\VENTLIQ\Output\MASSFLOW\MIXED\A')
.Value)*Factor;% Mass of acetone in the vent,g/min
```

```
BinVent=(Aspen.Application.Tree.FindNode('\Data\Streams\VENTLIQ\Output\MASSFLOW\MIXED\B')
.Value)*Factor;% Mass of acetone in vent,g/min

EinVent=(Aspen.Application.Tree.FindNode('\Data\Streams\VENTLIQ\Output\MASSFLOW\MIXED\ETO
H').Value)*Factor;% Mass of acetone in vent,g/min
  VABEVent =
Aspen.Application.Tree.FindNode('\Data\Streams\VENTLIQ\Output\VOLFLMX\MIXED').Value ;
%Volume of stripped stream, L

  ABEmassproduced =(AinReactor+BinReactor+EinReactor+AinVent+BinVent+EinVent); % Total
mass of ABE produced in g
  ABEProduced = ABEmassproduced/(VLIQS); % ABE productivity in g/L/h

  ABEmassinVent =(AinVent+BinVent+EinVent); % mass of ABE in vent, g/min
  %Total mass of acetone,butanol, ethanol, acetic acid, butyric acid, water
  %in the vent, g/min

TotalmassallLIQcompinVent=(Aspen.Application.Tree.FindNode('\Data\Streams\VENTLIQ\Output\
MASSFLMX\MIXED').Value)*Factor;

  ABEmassfracinVent= ABEmassinVent/TotalmassallLIQcompinVent; %mass fraction of ABE in
the vent, g/min
  y = ABEmassfracinVent;

  %Mass of ABE in the reactor, g/min
  ABEmassinRea = AinReactor+ BinReactor+ EinReactor;
  %Total mass of acetone,butanol, ethanol, acetic acid, butyric acid,
  %water, glucose in the reactor, g/min

TotalmassallLIQcompinReac=(Aspen.Application.Tree.FindNode('\Data\Streams\REALIQ\Output\M
ASSFLMX\MIXED').Value)*Factor;

  % Mass fraction of ABE in the reactor
  ABEmassfracinReac= ABEmassinRea/TotalmassallLIQcompinReac;
  xx = ABEmassfracinReac;

  % ABE selectivity
  ABESelectivity = y*(1-xx)/(xx*(1-y));
  ABEVentgperL = ABEmassinVent/VABEVent;

  % AAA is the objective function values ( the output of the objfunction)
  % determined at the current pair of decision variables
    AAA=-[ABEVentgperL  ABESelectivity ABEProduced]

 %Release COM object
  Aspen.Quit();
  delete(Aspen) % delete = Remove COM control or server

      end
  end
```

O.2: Output function for MOO code (visualize Pareto front and population changes)

The output function helps to visualize the Pareto front and population changes over successive generations.

```
function [state, options,optchanged] = outputfunction(options,state,flag)
persistent history
%displays the function eval value at each iteration. You can change this
disp(state.FunEval)
disp(state.Population)
disp(state.Rank)
disp(state.Selection)
figure
plot(state.Population(:,1)./0.065832,state.Population(:,2),'ro');
xlabel('Gas flowrate, L/min per L of broth')
ylabel('Gas stripping start time, h')
Numberofgenerations=state.Generation;
gen=sprintf('Number of generations = %d ', Numberofgenerations);
```

```
title(gen)
optchanged = false;

switch flag
 case 'init'
        disp('Starting the algorithm');
        history(:,:,1) = state.Population;
        assignin('base','gapopulationhistory',history);
    case {'iter','interrupt'}
        disp('Iterating ...')
        % Update the history every 1 generations.
        if rem(state.Generation,1) == 0
            ss = size(history,3);
            history(:,:,ss+1) = state.Population;
            assignin('base','gapopulationhistory',history);
        end
    case 'done'
        disp('Performing final task');
        % Include the final population in the history.
        ss = size(history,3);
        history(:,:,ss+1) = state.Population;
        assignin('base','gapopulationhistory',history);

end
```

O.3: SOO MATLAB code

```
function main

clc;clear; close all
set(0,'defaultAxesFontSize', 14)
set(0,'defaultlinelinewidth',3)

opts =
optimoptions(@ga,'PlotFcn',{@gaplotdistance,@gaplotselection,@gaplotscorediversity,@gaplo
tscores,@gaplotstopping,@gaplotbestf,@gaplotbestindiv,@gaplotexpectation,@gaplotrange},'F
unctionTolerance',0.01,'MaxGenerations',30,'display', 'iter',...
    'PopulationSize',500,'StallGenLimit',2)

rng default % for reproducibility % Random number generator

tic
% Call the function  'Calculation' that calculates the optimal values
[Decison_Variables,Objective_Function,eflag,outpt,population, score] = Calculation(opts)
toc

% Get a list of the final set of decision variables
x1=Decison_Variables(:,1)./0.06583 % Gas flow rate in kg/hr converted to L/min
x2=Decison_Variables(:,2)% Gas stripping start time, h

% Get the objective values
Objective_1 = -Objective_Function % Weighted average sum of three objective functions

End


% Function Calculation takes the input opts to give the decision variables
% (x), objective function values (f) etc.

function [x,f,eflag,outpt,pop,score] = Calculation(opts)

%Fun calls the function objfun to evaluate the objective function values
%for each pair of decision values generated within the range of the
%decision values specified
fun = @objfun; % the objective function, nested below

%lower and upper bounds of the decsion variables
lb = [0.006583 15]; % Gas stripping time started after 15 to 25 hrs of fermentation
ub = [0.32915 25]; % Gas flowrate of N2 0.1 to 5 L/min per L of broth converted
% x1 is gas flow rate and x2 is gas stripping start time
% ga uses the function fun and the in-built solver to determine the
% Pareto front and the optimal values.
tic
[x,f,eflag,outpt,pop,score] = ga(fun,2,[],[],[],[],lb,ub,[],opts)
toc

 function CCC =objfun(x)
 attempt = 1;  % Preallocate the first attempt
 trying = true; %Set while to true
 % The try catch loop runs the code in the try block until an error
 % occurs,in which case the code passes the excecution to the catch block.
 % The catch block gives information on what type of error there is through
 % ME and gives instructions on how to handle the error until it is
 % rectified
 while trying
    try
     BBB=@objfunction; % Set a handle to call the objective function to run normally
     CCC=BBB(x); % The objective function values are evaluated at the decision values (x)
     trying = false; % If there is an error and trying is set to false.

    catch ME
    disp(ME) %Displays the error
    fprintf('Failed attempt (%d). Retyring...',attempt)
    pause(1)
     attempt = attempt+1; % Sets attempt to 2 and objfunj to rerun the decision value
                          % pair of points that run into error.
    end

 end
    end


    function AAA = objfunction(x)
    format long
% Print out the current pair of decision values being evaluated for the objective
fucntion.
input =x

% Calculation of the objective functions

AspenVersion = 'apwn.document.35.0'; % Programme ID (Progid) of Aspen Plus document
class.
% Also 'apwn.document.34' to specify V8.8 as the version to be used.
%V9 is internal version 35, so you should use 35 instead of 34 for the document class

AspenPath = 'C:\Users\kda228\Desktop\RBatchtoMATLAB\votruba.bkp'; % Replace with the
simulation path where file  Aspen Plus file is located
AspenVisible = 1; % 0 for not visible (Make Aspen Plus file visible when MATLAB code is
run)
AspenDialogs = 1; % 0 for not visible (Make Aspen Plus Diagnositic page visible when
MATLAB code is run)

% Get pointer and load Aspen Plus comserver:
% actxserver = creates and manipulates objects from Matlab thar are exposed
% in an application that supports automation
Aspen = actxserver(AspenVersion);
% Aspen = a handle to the default interface of the server (actxserver)

% get = Used to Get property value from interface, or display properties
get(Aspen)

% Open the Aspen-Simulation
invoke(Aspen,'InitFromArchive2',AspenPath);

Aspen.visible = AspenVisible; % Make it visible
Aspen.SuppressDialogs = AspenDialogs; % Make simulation run messages visible

  % Set new values of the decision value
  NewGasflowmassrate= x(1); % Gas flow rate in kg/h
  NewStartTime = (x(2));   %Gas stripping start time h
  % Use the new gas stripping start time to set up the times for Aspen Plus
  Newasfeedtime1=(NewStartTime-0.001); % Substract 0.0001 from the time as the first time
slot, h- Running into issues, changed to 0.001
  Newasfeedtime2=NewStartTime; % Actual gas stripping start time, h
  Gasflowrate = (NewGasflowmassrate*1000/(60*1.097190232)); % Gas flow rate in L/min

  % Set the gas flow rate on the continous feed at the secont gas stripping
  % start time to the new mass flow rate of the gas in kg/h
  Aspen.Application.Tree.FindNode('\Data\Blocks\RBATCH1\Input\FPROF_FLOW\GAS\#1').Value =
NewGasflowmassrate;

  % Read the old gas stripping start time from the previous run
  oldtime1=
Aspen.Application.Tree.FindNode('\Data\Blocks\RBATCH1\Input\FPROF_TIME\GAS\#0').Value;

  %Compare the old old stripping start time from the previous run and the
  %new gas stripping start time to determine whether to write first to the first
  %time or second slot.
  if NewStartTime >oldtime1
  Aspen.Application.Tree.FindNode('\Data\Blocks\RBATCH1\Input\FPROF_TIME\GAS\#1').Value =
Newasfeedtime2; % Write 2nd time slot first
  Aspen.Application.Tree.FindNode('\Data\Blocks\RBATCH1\Input\FPROF_TIME\GAS\#0').Value =
Newasfeedtime1; %Write 1st time slot second
  else
  Aspen.Application.Tree.FindNode('\Data\Blocks\RBATCH1\Input\FPROF_TIME\GAS\#0').Value =
Newasfeedtime1; % write 1st time slot first
  Aspen.Application.Tree.FindNode('\Data\Blocks\RBATCH1\Input\FPROF_TIME\GAS\#1').Value =
Newasfeedtime2; %Write 2nd time slot second
  end

  % Set the new gas flow rate to the gas flow rate from the current pair of
  % point in L/min

Aspen.Application.Tree.FindNode('\Data\Streams\GAS\Input\TOTFLOW\MIXED').Value=Gasflowrat
e; % New Feed mass fraction of Ethanol

 %Reinit and run simulation
 Aspen.Reinit();  %
 %Reinitialize the simulation case. To reinitialize specific blocks or
%streams, use the Reinit member of the IHAPEngine class.

%Run the simulation case, asynchronously if the argument is True. If
%the user interface is visible, simulations should always be run
%asynchronously. Asynchronously allows the automation client to proceed with
%other tasks while waiting for the simulationrun to complete.
%Aspen Plus should always be run asynchronously if the application is visible.
Aspen.Run2();

% Save the results of the current run in Aspen Plus
Aspen.Save();
% Read the new data (output) results from Aspen Plus
  VLIQS=
Aspen.Application.Tree.FindNode('\Data\Blocks\RBATCH1\Output\USER\#3\32\VLIQS\LITERS').Va
lue;% Volume of the liquids in the reactor, Liters

 Factor=1000/60; %Convert mass from kg/hr to g/min;

AinReactor=(Aspen.Application.Tree.FindNode('\Data\Streams\REALIQ\Output\MASSFLOW\MIXED\A
').Value)*Factor; % Mass of acetone in reactor,g/min

BinReactor=(Aspen.Application.Tree.FindNode('\Data\Streams\REALIQ\Output\MASSFLOW\MIXED\B
').Value)*Factor; % Mass of butanol in reactor,g/min

EinReactor=(Aspen.Application.Tree.FindNode('\Data\Streams\REALIQ\Output\MASSFLOW\MIXED\E
TOH').Value)*Factor; % Mass of ethanol in reactor,g/min


AinVent=(Aspen.Application.Tree.FindNode('\Data\Streams\VENTLIQ\Output\MASSFLOW\MIXED\A')
.Value)*Factor;% Mass of acetone in the vent,g/min
```

```
BinVent=(Aspen.Application.Tree.FindNode('\Data\Streams\VENTLIQ\Output\MASSFLOW\MIXED\B')
.Value)*Factor;% Mass of acetone in vent,g/min

EinVent=(Aspen.Application.Tree.FindNode('\Data\Streams\VENTLIQ\Output\MASSFLOW\MIXED\ETO
H').Value)*Factor;% Mass of acetone in vent,g/min
  VABEVent =
Aspen.Application.Tree.FindNode('\Data\Streams\VENTLIQ\Output\VOLFLMX\MIXED').Value;

  ABEmassproduced =(AinReactor+BinReactor+EinReactor+AinVent+BinVent+EinVent); % Total
mass of ABE produced in g
  ABEProduced = ABEmassproduced/(VLIQS); % ABE productivity in g/L/h

  ABEmassinVent =(AinVent+BinVent+EinVent); % mass of ABE in vent, g/min
  %Total mass of acetone,butanol, ethanol, acetic acid, butyric acid, water
  %in the vent, g/min

TotalmassallLIQcompinVent=(Aspen.Application.Tree.FindNode('\Data\Streams\VENTLIQ\Output\
MASSFLMX\MIXED').Value)*Factor;

  ABEmassfracinVent= ABEmassinVent/TotalmassallLIQcompinVent; %mass fraction of ABE in
the vent, g/min
  y = ABEmassfracinVent;

  %Mass of ABE in the reactor, g/min
  ABEmassinRea = AinReactor+ BinReactor+ EinReactor;
  %Total mass of acetone,butanol, ethanol, acetic acid, butyric acid,
  %water, glucose in the reactor, g/min

TotalmassallLIQcompinReac=(Aspen.Application.Tree.FindNode('\Data\Streams\REALIQ\Output\M
ASSFLMX\MIXED').Value)*Factor;

  % Mass fraction of ABE in the reactor
  ABEmassfracinReac= ABEmassinRea/TotalmassallLIQcompinReac;
  xx = ABEmassfracinReac;

  % ABE selectivity
  ABESelectivity = y*(1-xx)/(xx*(1-y));
  ABEVentgperL = ABEmassinVent/VABEVent;
  % AAA is the objective function values ( the output of the objfunction)
  % determined at the current pair of decision variables

 AAA=-((1/3*ABEVentgperL)+(1/3*ABESelectivity)+(1/3* ABEProduced)) % Negative for
maximization

%Release COM object
  Aspen.Quit();
  delete(Aspen) % delete = Remove COM control or server


      end
  end
```

Appendix P: MOO for the batch ethanolic fermentation with in situ gas stripping used in the Chapter 6

P.1: Main MOO code

```
function main

clc;clear; close all
set(0,'defaultAxesFontSize', 32)
set(0,'defaultlinelinewidth',3)

opts =
optimoptions(@gamultiobj,'PlotFcn',{@gaplotpareto},'FunctionTolerance',0.01,'MaxGeneratio
ns',30,'display', 'iter',...    'OutputFcns',
@outputfunction,'PopulationSize',1000,'StallGenLimit',5)%,'InitialPopulation',Initpopu)%,
'UseParallel','always');%'UseVectorized',true); % default ga options settings

rng default % for reproducibility % Random number generator

tic
% Call the function  'Calculation' that calculates the optimal values
[Decison_Variables,Objective_Function,eflag,outpt,population, score] = Calculation(opts)
toc

% Get a list of the final set of decision variables
x1=Decison_Variables(:,1) % Initial substrate concentration, g/L
x2=Decison_Variables(:,2)% Batch fermentation time, h
x3=Decison_Variables(:,3)% Gas stripping start time, h
x4=Decison_Variables(:,4)./0.06583% % Gas flow rate in kg/hr converted to L/min

% Get the objective values
Objective_1 = -Objective_Function(:,1) % Ethanol Yield, g/g
Objective_2 = -Objective_Function(:,2) %Ethanol Productivity, g/L.h
Objective_3 = -Objective_Function(:,3) %Ethanol Produced, g/L
Objective_4 = -Objective_Function(:,4) %Total sugars conversion fraction

end

% Function Calculation takes the input opts to give the decision variables
% (x), objective function values (f) etc.
function [x,f,eflag,outpt,pop,score] = Calculation(opts)

%Fun calls the function objfun to evaluate the objective function values
%for each pair of decision values generated within the range of the
%decision values specified
fun = @objfun; % the objective function, nested below

%lower and upper bounds of the decsion variables
lb = [10 5 5 0.0006583]; % Initial sugar concentration, g/L;Batch fermentation time,
h;Gas stripping start time, h and Gas flow rate in kg/hr converted to L/min
ub = [300 100 100 0.19749];

%Inquality constraint: x3-x2<=0
ARHS=[0 -1 1 0];
BLHS = 0;
% gamultiobj uses the function fun and the in-built solver to determine the
% Pareto front and the optimal values.
tic
[x,f,eflag,outpt,pop,score] = gamultiobj(fun,4,ARHS,BLHS,[],[],lb,ub,[],opts)
toc

 function CCC =objfun(x)
 attempt = 1;  % Preallocate the first attempt
 trying = true; %Set while to true
 % The try catch loop runs the code in the try block until an error
 % occurs,in which case the code passes the excecution to the catch block.
 % The catch block gives information on what type of error there is through
 % ME and gives instructions on how to handle the error until it is
 % rectified
 while trying
    try
     BBB=@objfunction; % Set a handle to call the objective function to run normally
     CCC=BBB(x); % The objective function values are evaluated at the decision values (x)
     trying = false; % If there is an error and trying is set to false.

    catch ME
    disp(ME) %Displays the error
    fprintf('Failed attempt (%d). Retyring...',attempt)
    pause(1)
     attempt = attempt+1; % Sets attempt to 2 and objfunj to rerun the decision value
                          % pair of points that run into error.
    end

 end
    end

    function AAA = objfunction(x)

% Print out the current pair of decision values being evaluated for the objective
fucntion.
input =x

% Calculation of the objective functions

AspenVersion = 'apwn.document.35.0'; % Programme ID (Progid) of Aspen Plus document
class.
% Also 'apwn.document.34' to specify V8.8 as the version to be used.
%V9 is internal version 35, so you should use 35 instead of 34 for the document class

AspenPath = 'C:\Users\kda228\Desktop\ETHANOL_REMOVE_ETHANOL\batch.bkp'; % Replace with
the simulation path where file  Aspen Plus file is located
AspenVisible = 1; % 0 for not visible (Make Aspen Plus file visible when MATLAB code is
run)
AspenDialogs = 1; % 0 for not visible (Make Aspen Plus Diagnositic page visible when
MATLAB code is run)

% Get pointer and load Aspen Plus comserver:
% actxserver = creates and manipulates objects from Matlab thar are exposed
% in an application that supports automation
Aspen = actxserver(AspenVersion);
% Aspen = a handle to the default interface of the server (actxserver)

% get = Used to Get property value from interface, or display properties
get(Aspen)

% Open the Aspen-Simulation
invoke(Aspen,'InitFromFile2',AspenPath);

Aspen.visible = AspenVisible; % Make it visible
Aspen.SuppressDialogs = AspenDialogs; % Make simulation run messages visible

  Substrateconcentration= x(1); % Gas flow rate in kg/h
  FermentationTime = x(2);   %Gas stripping start time h

  % Set new values of the decision value
  NewGasflowmassrate= x(4); % Gas flow rate in kg/h
  NewStartTime = x(3);   %Gas stripping start time h

  % Use the new gas stripping start time to set up the times for Aspen Plus
  Newasfeedtime1=(NewStartTime-0.01); % Substract 0.001 from the time as the first time
slot, h- Running into issues of Aspen Plus integration crashing, changed to 0.01
  Newasfeedtime2=NewStartTime; % Actual gas stripping start time, h
  Gasflowrate = (NewGasflowmassrate*1000/(60*1.097190232)); % Gas flow rate in L/min

  % Set the gas flow rate on the continous feed at the secont gas stripping
  % start time to the new mass flow rate of the gas in kg/h
  Aspen.Application.Tree.FindNode('\Data\Blocks\RBATCH1\Input\FPROF_FLOW\GAS\#1').Value =
NewGasflowmassrate;

  % Read the old gas stripping start time from the previous run
  oldtime1=
Aspen.Application.Tree.FindNode('\Data\Blocks\RBATCH1\Input\FPROF_TIME\GAS\#0').Value;

  %Compare the old old stripping start time from the previous run and the
  %new gas stripping start time to determine whether to write first to the first
  %time or second slot.
  if NewStartTime >oldtime1
  Aspen.Application.Tree.FindNode('\Data\Blocks\RBATCH1\Input\FPROF_TIME\GAS\#1').Value =
Newasfeedtime2; % Write 2nd time slot first
  Aspen.Application.Tree.FindNode('\Data\Blocks\RBATCH1\Input\FPROF_TIME\GAS\#0').Value =
Newasfeedtime1; %Write 1st time slot second
  else
  Aspen.Application.Tree.FindNode('\Data\Blocks\RBATCH1\Input\FPROF_TIME\GAS\#0').Value =
Newasfeedtime1; % write 1st time slot first
  Aspen.Application.Tree.FindNode('\Data\Blocks\RBATCH1\Input\FPROF_TIME\GAS\#1').Value =
Newasfeedtime2; %Write 2nd time slot second
  end

  % Set the new gas flow rate to the gas flow rate from the current pair of
  % point in L/min

Aspen.Application.Tree.FindNode('\Data\Streams\GAS\Input\TOTFLOW\MIXED').Value=Gasflowrat
e; % New Feed mass fraction of Ethanol


  % Set fermentation times
  Aspen.Application.Tree.FindNode('\Data\Blocks\RBATCH1\Input\VALUE\1').Value =
FermentationTime; % Stopping time,h
   Aspen.Application.Tree.FindNode('\Data\Blocks\RBATCH1\Input\MAX_TIME').Value =
FermentationTime; % Maximum calculation time, h
  Aspen.Application.Tree.FindNode('\Data\Blocks\RBATCH1\Input\MAX_NPOINT').Value =
FermentationTime+2; %Maximum number of profile points

  % Set glucose and xylose concentrations
  Aspen.Application.Tree.FindNode('\Data\Streams\FEED1\Input\FLOW\MIXED\S').Value =
Substrateconcentration;
  Aspen.Application.Tree.FindNode('\Data\Streams\FEED1\Input\FLOW\MIXED\XYLOSE').Value =
Substrateconcentration;

  %Reinit and run simulation
 Aspen.Reinit();  %
 %Reinitialize the simulation case. To reinitialize specific blocks or


Aspen.Run2();
%Run the simulation case, asynchronously if the argument is True. If
%the user interface is visible, simulations should always be run
%asynchronously. Asynchronously allows the automation client to proceed with
%other tasks while waiting for the simulationrun to complete.
%Aspen Plus should always be run asynchronously if the application is visible.

% Save the results of the current run in Aspen Plus
Aspen.Save();

  % Read data
  %conversion of kg/hr to g/min (kg/h)*(1000g/1kg)*(1h/60min)
  Factor = (1000/60);
  InitialVolume=
Aspen.Application.Tree.FindNode('\Data\Streams\FEED1\Output\VOLFLMX\MIXED').Value;
%Volume of Initial mixture in reactor, L
  FinalVolume=
Aspen.Application.Tree.FindNode('\Data\Streams\RELIQ\Output\VOLFLMX\MIXED').Value;
%Volume of final mixture in reactor, L

VentVolume=Aspen.Application.Tree.FindNode('\Data\Streams\VENTLIQ\Output\VOLFLMX\MIXED').
Value; %Volume of final mixture in stripped stream, L

EthanolMassReactor=(Aspen.Application.Tree.FindNode('\Data\Streams\RELIQ\Output\MASSFLOW\
MIXED\ETOH').Value)*Factor; % g/L
```

```
  EthanolMassVent =
(Aspen.Application.Tree.FindNode('\Data\Streams\VENTLIQ\Output\MASSFLOW\MIXED\ETOH').Valu
e)*Factor; % g/L


InitialEthanolMass=(Aspen.Application.Tree.FindNode('\Data\Streams\FEED1\Input\FLOW\MIXED
\ETOH').Value)*InitialVolume; % g/L
  Ethanolproduced= (EthanolMassReactor+EthanolMassVent-
InitialEthanolMass)/(FinalVolume+VentVolume); %g/L
  MassofEthanolProduced =(EthanolMassReactor+EthanolMassVent-InitialEthanolMass);

  InitialMassGlucose=
(Aspen.Application.Tree.FindNode('\Data\Streams\FEED1\Input\FLOW\MIXED\S').Value)*(Initia
lVolume);
  FinalMassGlucose=
(Aspen.Application.Tree.FindNode('\Data\Streams\PRODUCT1\Output\MASSFLOW\MIXED\S').Value)
*(FinalVolume)*Factor;
  GlucoseUsed = InitialMassGlucose-FinalMassGlucose;

  InitialMassXylose=
(Aspen.Application.Tree.FindNode('\Data\Streams\FEED1\Input\FLOW\MIXED\XYLOSE').Value)*(I
nitialVolume);
  FinalMassXylose=
(Aspen.Application.Tree.FindNode('\Data\Streams\PRODUCT1\Output\MASSFLOW\MIXED\XYLOSE').V
alue)*(FinalVolume)*Factor;
  XyloseUsed = InitialMassXylose-FinalMassXylose;

  EthanolYield=   MassofEthanolProduced/(GlucoseUsed+XyloseUsed);

  EthanolProductivity=Ethanolproduced/FermentationTime;
  SugarConversionFraction = (GlucoseUsed+
XyloseUsed)/(InitialMassGlucose+InitialMassXylose);

  EthanolCondensate=EthanolMassVent/VentVolume


Ethanolmassfractionreactor=(Aspen.Application.Tree.FindNode('\Data\Streams\RELIQ\Output\M
ASSFRAC\MIXED\ETOH').Value);

Ethanolmassfractionvent=(Aspen.Application.Tree.FindNode('\Data\Streams\VENTLIQ\Output\MA
SSFRAC\MIXED\ETOH').Value);

  xx=Ethanolmassfractionreactor;
  yy=Ethanolmassfractionvent;
  EthanolSelectivity =yy*(1-xx)/(xx*(1-yy))
  AAA=-[EthanolYield  EthanolProductivity  Ethanolproduced SugarConversionFraction]


%Release COM object
  Aspen.Quit();
  delete(Aspen) % delete = Remove COM control or server


      end
  end
```

P.2: Output function for MOO code

```
function [state, options,optchanged] = outputfunction(options,state,flag)
persistent history
%displays the function eval value at each iteration. You can change this
disp(state.FunEval)
disp(state.Population)
disp(state.Rank)
disp(state.Selection)
figure
plot(state.Population(:,1),state.Population(:,2),'ro');
xlabel('Initial Substrate Concentration, g/L')
ylabel('Fermentation time, h')
Numberofgenerations=state.Generation;
gen=sprintf('Number of generations = %d ', Numberofgenerations);
```

```
title(gen)

figure
plot(state.Population(:,1),state.Population(:,3),'ro');
xlabel('Initial Substrate Concentration, g/L')
ylabel('Stripping time, h')
Numberofgenerations=state.Generation;
gen=sprintf('Number of generations = %d ', Numberofgenerations);
title(gen)

figure
plot(state.Population(:,1),state.Population(:,4)./0.06583,'ro');
xlabel('Initial Substrate Concentration, g/L')
ylabel('Gas flow rate, L/min per L')
Numberofgenerations=state.Generation;
gen=sprintf('Number of generations = %d ', Numberofgenerations);
title(gen)

figure
plot(state.Population(:,2),state.Population(:,3),'ro');
xlabel('Fermentation time, h')
ylabel('Stripping time, h')
Numberofgenerations=state.Generation;
gen=sprintf('Number of generations = %d ', Numberofgenerations);
title(gen)

figure
plot(state.Population(:,2),state.Population(:,4)./0.06583,'ro');
xlabel('Fermentation time, h')
ylabel('Gas flow rate, L/min per L')
Numberofgenerations=state.Generation;
gen=sprintf('Number of generations = %d ', Numberofgenerations);
title(gen)

figure
plot(state.Population(:,3),state.Population(:,4)./0.06583,'ro');
xlabel('Stripping time, h')
ylabel('Gas flow rate, L/min per L')
Numberofgenerations=state.Generation;
gen=sprintf('Number of generations = %d ', Numberofgenerations);
title(gen)
optchanged = false;

switch flag
 case 'init'
        disp('Starting the algorithm');
        history(:,:,1) = state.Population;
        assignin('base','gapopulationhistory',history);
    case {'iter','interrupt'}
        disp('Iterating ...')
        % Update the history every 1 generations.
        if rem(state.Generation,1) == 0
            ss = size(history,3);
            history(:,:,ss+1) = state.Population;
            assignin('base','gapopulationhistory',history);
        end
    case 'done'
        disp('Performing final task');
        % Include the final population in the history.
        ss = size(history,3);
        history(:,:,ss+1) = state.Population;
        assignin('base','gapopulationhistory',history);

end
```

References

1. Ezeji, T. C.; Qureshi, N.; Blaschek, H., Microbial production of a biofuel (acetone–butanol–ethanol) in a continuous bioreactor: impact of bleed and simultaneous product removal. *Bioprocess Biosyst. Eng.* **2013,** *36* (1), 109-116.

2. Mears, L.; Stocks, S. M.; Albaek, M. O.; Sin, G.; Gernaey, K. V., Mechanistic Fermentation Models for Process Design, Monitoring, and Control. *Trends Biotechnol.* **2017**.

3. Lin, Y.; Tanaka, S., Ethanol fermentation from biomass resources: current state and prospects. *Appl. Microbiol. Biotechnol.* **2006,** *69* (6), 627-642.

4. Dürre, P., Fermentative Butanol Production. *Ann. N.Y. Acad. Sci.* **2008,** *1125* (1), 353-362.

5. Van der Merwe, A. B. Evaluation of different process designs for biobutanol production from sugarcane molasses. M. S. Thesis, University of Stellenbosch, Stellenbosch, South Africa, 2010.

6. Shuler, M. L.; Kargi, F., *Bioprocess Engineering Basic Concepts*. 2nd ed.; Prentice hall: New Jersey, 2009.

7. Vane, L. M., Separation technologies for the recovery and dehydration of alcohols from fermentation broths. *Biofuels, Bioprod. Biorefin.* **2008,** *2* (6), 553-588.

8. Thatipamala, R.; Rohani, S.; Hill, G., Effects of high product and substrate inhibitions on the kinetics and biomass and product yields during ethanol batch fermentation. *Biotechnol. Bioeng.* **1992,** *40* (2), 289-297.

9. Lee, K. J.; Rogers, P. L., The fermentation kinetics of ethanol production by Zymomonas mobilis. *The Chemical Engineering Journal* **1983,** *27* (2), B31-B38.

10. Ezeji, T. C.; Qureshi, N.; Blaschek, H. P., Acetone butanol ethanol (ABE) production from concentrated substrate: reduction in substrate inhibition by fed-batch technique and product inhibition by gas stripping. *Appl. Microbiol. Biotechnol.* **2004,** *63* (6), 653-658.

11. Qureshi, N.; Blaschek, H., Recent advances in ABE fermentation: hyper-butanol producing Clostridium beijerinckii BA101. *J. Ind. Microbiol. Biotechnol.* **2001,** *27* (5), 287-291.

12. Lassmann, T.; Kravanja, P.; Friedl, A., Simulation of the downstream processing in the ethanol production from lignocellulosic biomass with ASPEN Plus® and IPSEpro. *Energy, Sustainability and Society* **2014,** *4* (1), 27.

13. DOE, U. S. *Lignocellulosic biomass for advanced biofuels and bioproducts: Workshop report*; DOE/SC-0170.; 2015.

14. Huang, H. J.; Ramaswamy, S., Overview of biomass conversion processes and separation and purification technologies in biorefineries. *Separation and Purification Technologies in Biorefineries* **2013**, 1-36.

15. Lee, J., Biological conversion of lignocellulosic biomass to ethanol. *J. Biotechnol.* **1997,** *56* (1), 1-24.

16. Dussan, K. J.; Silva, D.; Moraes, E.; Arruda, P. V.; Felipe, M., Dilute-acid hydrolysis of cellulose to glucose from sugarcane bagasse. *Chemical Engineering Transaction* **2014,** *38*.

17. Abels, C.; Thimm, K.; Wulfhorst, H.; Spiess, A. C.; Wessling, M., Membrane-based recovery of glucose from enzymatic hydrolysis of ionic liquid pretreated cellulose. *Bioresour Technol* **2013,** *149*, 58-64.

18. Malmali, M.; Stickel, J. J.; Wickramasinghe, S. R., Sugar concentration and detoxification of clarified biomass hydrolysate by nanofiltration. *Separation and Purification Technology* **2014,** *132*, 655-665.

19. Song, Y.; Wi, S. G.; Kim, H. M.; Bae, H.-J., Cellulosic bioethanol production from Jerusalem artichoke (Helianthus tuberosus L.) using hydrogen peroxide-acetic acid (HPAC) pretreatment. *Bioresour. Technol.* **2016,** *214*, 30-36.

20. Kiss, A. A.; Grievink, J.; Rito-Palomares, M., A systems engineering perspective on process integration in industrial biotechnology. *J. Chem. Technol. Biotechnol.* **2015,** *90* (3), 349-355.

21. Stephanopoulos, G.; Reklaitis, G. V., Process systems engineering: From Solvay to modern bio-and nanotechnology.: A history of development, successes and prospects for the future. *Chem. Eng. Sci.* **2011,** *66* (19), 4272-4306.

22. Sharma, P.; Sarker, B.; Romagnoli, J. A., A decision support tool for strategic planning of sustainable biorefineries. *Comput. Chem. Eng.* **2011,** *35* (9), 1767-1781.

23. Laurens, L. M.; Markham, J.; Templeton, D. W.; Christensen, E. D.; Van Wychen, S.; Vadelius, E. W.; Chen-Glasser, M.; Dong, T.; Davis, R.; Pienkos, P. T., Development of algae biorefinery concepts for biofuels and bioproducts; a perspective on process-compatible products and their impact on cost-reduction. *Energy Environ. Sci.* **2017,** *10* (8), 1716-1738.

24. Black, M. J.; Sadhukhan, J.; Day, K.; Drage, G.; Murphy, R. J., Developing database criteria for the assessment of biomass supply chains for biorefinery development. *Chem. Eng. Res. Des.* **2016,** *107* (Supplement C), 253-262.

25. Zhu, L., Biorefinery as a promising approach to promote microalgae industry: An innovative framework. *Renew. Sustainable Energy Rev.* **2015,** *41*, 1376-1384.

26. Soh, L.; Eckelman, M. J., Green solvents in biomass processing. *ACS Sustain. Chem. Eng.* **2016,** *4* (11), 5821-5837.

27. Gudena, K.; Rangaiah, G. P.; Lakshminarayanan, S., *Multi-Objective Optimization in Chemical Engineering: Developments and Applications*. First ed.; John Wiley & Sons, Ltd.: United Kingdom, 2013.

28. Thaibault, J., *Net Flow and Rough Sets: Two Methods for Ranking the Pareto Domain*. First ed.; World Scientific Publishing Co. Pte. Ltd.: Singapore, 2009; Vol. 1, p 1-3;189-235.

29. Srinivasan, B.; Palanki, S.; Bonvin, D., Dynamic optimization of batch processes: I. Characterization of the nominal solution. *Comput. Chem. Eng.* **2003,** *27* (1), 1-26.

30. Gosling, I., Process simulation and modeling for industrial bioprocessing: Tools and techniques. *Ind. Biotechnol.* **2005,** *1* (2), 106-109.

31. Alnur, A.; Sakinah, M.; Mustafa, A.-B.; Kamarudin, H.; Norazian, M., Simulation of xylitol production: a review. *Aust. J. Basic Appl. Sci.* **2013,** *7* (5), 366-372.

32. Petrides, D.; Carmichael, D.; Siletti, C.; Koulouris, A., Biopharmaceutical process optimization with simulation and scheduling tools. *Bioeng.* **2014,** *1* (4), 154-187.

33. Bonvin, D.; Georgakis, C.; Pantelides, C.; Barolo, M.; Grover, M.; Rodrigues, D.; Schneider, R.; Dochain, D., Linking models and experiments. *Ind. Eng. Chem. Res.* **2016,** *55* (25), 6891-6903.

34. Xia, J.; Wang, G.; Lin, J.; Wang, Y.; Chu, J.; Zhuang, Y.; Zhang, S., Advances and practices of bioprocess scale-up. In *Bioreactor Engineering Research and Industrial Applications II*, Springer: 2015; pp 137-151.

35. Koutinas, M.; Kiparissides, A.; Pistikopoulos, E. N.; Mantalaris, A., Bioprocess systems engineering: transferring traditional process engineering principles to industrial biotechnology. *Comput. Struct. Biotechnol. J.* **2012,** *3* (4), 1-9.
36. Kostov, G.; Popova, S.; Gochev, V.; Koprinkova-Hristova, P.; Angelov, M.; Georgieva, A., Modeling of batch alcohol fermentation with free and immobilized yeasts Saccharomyces cerevisiae 46 EVD. *Biotechnol. Biotechnol. Equip.* **2012,** *26* (3), 3021-3030.
37. Nigam, P. S.; Singh, A., Production of liquid biofuels from renewable resources. *Prog. Energy Combust. Sci.* **2011,** *37* (1), 52-68.
38. Abdehagh, N.; Tezel, F. H.; Thibault, J., Separation techniques in butanol production: Challenges and developments. *Biomass Bioenergy* **2014,** *60* (0), 222-246.
39. Yue, D.; You, F.; Snyder, S. W., Biomass-to-bioenergy and biofuel supply chain optimization: Overview, key issues and challenges. *Comput. Chem. Eng.* **2014,** *66* (0), 36-56.
40. Seay, J. R.; Badurdeen, F. F., Current trends and directions in achieving sustainability in the biofuel and bioenergy supply chain. *Current Opinion in Chemical Engineering* **2014,** *6* (0), 55-60.
41. Qureshi, N.; Singh, V.; Liu, S.; Ezeji, T. C.; Saha, B. C.; Cotta, M. A., Process integration for simultaneous saccharification, fermentation, and recovery (SSFR): Production of butanol from corn stover using Clostridium beijerinckii P260. *Bioresour. Technol.* **2014,** *154* (0), 222-228.
42. Ezeji, T. C.; Qureshi, N.; Blaschek, H. P., Production of acetone, butanol and ethanol by Clostridium beijerinckii BA101 and in situ recovery by gas stripping. *World J. Microbiol. Biotechnol.* **2003,** *19* (6), 595-603.
43. Sukumara, S.; Faulkner, W.; Amundson, J.; Badurdeen, F.; Seay, J., A multidisciplinary decision support tool for evaluating multiple biorefinery conversion technologies and supply chain performance. *Clean Technol. Environ. Policy* **2014,** *16* (6), 1027-1044.
44. de Jong, E.; Jungmeier, G., Biorefinery concepts in comparison to petrochemical refineries. *Industrial Biorefineries & White Biotechnology* **2015**, 3-33.
45. Sukumara, S. A multidisciplinary techno-economic decision support tool for validating long-term economic viabiity of biorefining processes. University of Kentucky, Lexington, Kentucky, 2014.
46. Lee, R. A.; Lavoie, J.-M., From first-to third-generation biofuels: Challenges of producing a commodity from a biomass of increasing complexity. *Animal Frontiers* **2013,** *3* (2), 6-11.
47. Naik, S. N.; Goud, V. V.; Rout, P. K.; Dalai, A. K., Production of first and second generation biofuels: A comprehensive review. *Renewable and Sustainable Energy Reviews* **2010,** *14* (2), 578-597.
48. Lu, J.; Sheahan, C.; Fu, P., Metabolic engineering of algae for fourth generation biofuels production. *Energy & Environmental Science* **2011,** *4* (7), 2451-2466.
49. Wald, S.; Wilke, C. R.; Blanch, H. W., Kinetics of the enzymatic hydrolysis of cellulose. *Biotechnol. Bioeng.* **1984,** *26* (3), 221-230.
50. Jacobsen, S.; Wyman, C., Cellulose and hemicellulose hydrolysis models for application to current and novel pretreatment processes. *Appl. Biochem. Biotechnol.* **2000,** *84-86* (1-9), 81-96.

51. Tripp, S.; Powell, R.; Nelson, P., Regional strategy for biobased products in the Mississippi Delta. *Report by Battelle Technology Partnership Practice* **2009**.
52. Geraili, A.; Sharma, P.; Romagnoli, J. A., Technology analysis of integrated biorefineries through process simulation and hybrid optimization. *Energy* **2014,** *73* (0), 145-159.
53. Kumar, P.; Barrett, D. M.; Delwiche, M. J.; Stroeve, P., Methods for pretreatment of lignocellulosic biomass for efficient hydrolysis and biofuel production. *Industrial & Engineering Chemistry Research* **2009,** *48* (8), 3713-3729.
54. Taherzadeh, M. J.; Karimi, K., Enzymatic-based hydrolysis processes for ethanol. *BioResources* **2007,** *2* (4), 707-738.
55. Fontes, C. M.; Gilbert, H. J., Cellulosomes: highly efficient nanomachines designed to deconstruct plant cell wall complex carbohydrates. *Annu. Rev. Biochem* **2010,** *79*, 655-681.
56. Lynd, L. R.; Weimer, P. J.; Van Zyl, W. H.; Pretorius, I. S., Microbial cellulose utilization: fundamentals and biotechnology. *Microbiology and molecular biology reviews* **2002,** *66* (3), 506-577.
57. Demain, A. L.; Newcomb, M.; Wu, J. D., Cellulase, clostridia, and ethanol. *Microbiology and molecular biology reviews* **2005,** *69* (1), 124-154.
58. Sun, Y.; Cheng, J., Hydrolysis of lignocellulosic materials for ethanol production: a review. *Bioresour. Technol.* **2002,** *83* (1), 1-11.
59. Kumar, R.; Tabatabaei, M.; Karimi, K.; Sárvári Horváth, I., Recent updates on lignocellulosic biomass derived ethanol-A review. *Biofuel Research Journal* **2016,** *3* (1), 347-356.
60. Montoya, D.; Spitia, S.; Silva, E.; Schwarz, W. H., Isolation of mesophilic solvent-producing clostridia from Colombian sources: physiological characterization, solvent production and polysaccharide hydrolysis. *J. Biotechnol.* **2000,** *79* (2), 117-126.
61. Ezeji, T. C.; Qureshi, N.; Blaschek, H. P., Bioproduction of butanol from biomass: from genes to bioreactors. *Curr. Opin. Biotechnol.* **2007,** *18* (3), 220-227.
62. Seader, J. D.; Henley, E. J., *Separation Process Principles*. John Wiley & SOns, Inc.: 2006.
63. Cussler, E. L., *Diffusion: Mass transfer in fluid systems*. Third ed.; Cambridge University Press: New York, 2012.
64. Qureshi, N.; Hughes, S.; Maddox, I. S.; Cotta, M. A., Energy-efficient recovery of butanol from model solutions and fermentation broth by adsorption. *Bioprocess Biosyst. Eng.* **2005,** *27* (4), 215-222.
65. BP Statistical Review of World Energy. https://www.bp.com/content/dam/bp/en/corporate/pdf/energy-economics/statistical-review-2017/bp-statistical-review-of-world-energy-2017-full-report.pdf (accessed Oct 12, 2017).
66. van Eijck, J.; Batidzirai, B.; Faaij, A., Current and future economic performance of first and second generation biofuels in developing countries. *Applied Energy* **2014,** *135*, 115-141.
67. Chen, Z.; Andresen, S., A multiobjective optimization model of production-sourcing for sustainable supply chain with consideration of social, environmental, and economic factors. *Math. Probl. Eng.* **2014,** *2014*, doi:10.1155/2014/616107.
68. Seay, J. R., Education for sustainability: Developing a taxonomy of the key principles for sustainable process and product design. *Comput. Chem. Eng.* **2015,** *81*, 147-152.

69. Geraili, A.; Sharma, P.; Romagnoli, J. A., A modeling framework for design of nonlinear renewable energy systems through integrated simulation modeling and metaheuristic optimization: Applications to biorefineries. *Comput. Chem. Eng.* **2014,** *61* (0), 102-117.
70. Darkwah, K.; Wang, L.; Shahbazi, A., Simultaneous saccharification and fermentation of sweet sorghum after acid pretreatment. *Energy Sources, Part A: Recovery, Utilization, and Environmental Effects* **2016,** *38* (10), 1485-1492.
71. E4tech; RE-CORD; WUR. *Final Report for the European Commission*; ENER/C2/423-2012/SI2.673791; 2015.
72. Chinn, M. S.; Nokes, S. E.; Strobel, H. J., Influence of process conditions on end product formation from Clostridium thermocellum 27405 in solid substrate cultivation on paper pulp sludge. *Bioresour. Technol.* **2007,** *98* (11), 2184-2193.
73. Klatt, K.-U.; Marquardt, W., Perspectives for process systems engineering—Personal views from academia and industry. *Comput. Chem. Eng.* **2009,** *33* (3), 536-550.
74. Grossmann, I. E.; Westerberg, A. W., Research challenges in process systems engineering. *AlChE J.* **2000,** *46* (9), 1700-1703.
75. Jiménez-González, C.; Woodley, J. M., Bioprocesses: Modeling needs for process evaluation and sustainability assessment. *Comput. Chem. Eng.* **2010,** *34* (7), 1009-1017.
76. Shanklin, T.; Roper, K.; Yegneswaran, P.; Marten, M. R., Selection of bioprocess simulation software for industrial applications. *Biotechnol. Bioeng.* **2001,** (72), 483-489.
77. Hukkerikar, A.; Jones, M. N.; Sarup, B.; Abildskov, J.; Sin, G.; Gani, R. In *11th International Symposium on Process Systems Engineering-PSE2012*, Sensitivity of Process Design due to Uncertainties in Property Estimates, Elsevier: Sensitivity of Process Design due to Uncertainties in Property Estimates, 2012; p 200.
78. Petrides, D.; Cooney, C. L.; Evans, L. B.; Field, R. P.; Snoswell, M., Bioprocess simulation: an integrated approach to process development. *Comput. Chem. Eng.* **1989,** *13* (4-5), 553-561.
79. Carlson, E. C., Don't gamble with physical properties for simulations. *Chem. Eng. Prog.* **1996,** *92* (10), 35-46.
80. Ng, L. Y.; Chong, F. K.; Chemmangattuvalappil, N. G. In *Challenges and opportunities in computer aided molecular design*, Proceedings of the 8th International Conference on Foundations of Computer-Aided Process Design, Elsevier: 2014; p 25.
81. Hill, D.; Justice, F. C., Understand thermodynamics to improve process simulations. *Chem. Eng. Prog.* **2011,** *107*, 20-25.
82. You, F.; Tao, L.; Graziano, D. J.; Snyder, S. W., Optimal design of sustainable cellulosic biofuel supply chains: multiobjective optimization coupled with life cycle assessment and input–output analysis. *AlChE J.* **2012,** *58* (4), 1157-1180.
83. Vázquez-Ojeda, M.; Segovia-Hernández, J. G.; Hernández, S.; Hernández-Aguirre, A.; Kiss, A. A., Design and optimization of an ethanol dehydration process using stochastic methods. *Sep. Purif. Technol.* **2013,** *105*, 90-97.
84. Sharif Rohani, A.; Mehrani, P.; Thibault, J., Comparison of in-situ recovery methods of gas stripping, pervaporation, and vacuum separation by multi-objective optimization for producing biobutanol via fermentation process. *Can. J. Chem. Eng.* **2015,** *93* (6), 986-997.
85. Rohani, A. S. Multi-objective optimization of butanol production during ABE fermentation. M.S. Thesis, University of Ottawa, Ottawa, Canada, 2013.

86. Mesfun, S.; Lundgren, J.; Grip, C.-E.; Toffolo, A.; Rova, U. In *Techno-economic evaluation of butanol production via black liquor fractionation*, Proc. 12th Int. Conf. Sustain. energy Technol., Hong Kong: Faculty of Construction and Environment & Research Institute for Sutainable Urban Development, The Hong Kong Polytechnic University, China, 2013; pp 1639-1650.
87. Wu, M.; Wang, M.; Liu, J.; Huo, H., Life-Cycle Assessment of Corn-Based Butanol as a Potential Transportation Fuel. **2007**.
88. Aden, A.; Ruth, M.; Ibsen, K.; Jechura, J.; Neeves, K.; Sheehan, J.; Wallace, B.; Montague, L.; Slayton, A.; Lukas, J. *Lignocellulosic biomass to ethanol process design and economics utilizing co-current dilute acid prehydrolysis and enzymatic hydrolysis for corn stover*; No. NREL/TP-510-32438; NATIONAL RENEWABLE ENERGY LAB GOLDEN CO: 2002.
89. Humbird, D.; Davis, R.; Tao, L.; Kinchin, C.; Hsu, D.; Aden, A.; Schoen, P.; Lukas, J.; Olthof, B.; Worley, M. *Process design and economics for biochemical conversion of lignocellulosic biomass to ethanol: dilute-acid pretreatment and enzymatic hydrolysis of corn stover*; No. NREL/TP-5100-47764; National Renewable Energy Laboratory (NREL), Golden, CO.: 2011.
90. Quintero, J. n. A.; Cardona, C. A., Process simulation of fuel ethanol production from lignocellulosics using Aspen Plus. *Ind. Eng. Chem. Res.* **2011,** *50* (10), 6205-6212.
91. Dias, M. O.; Cunha, M. P.; Jesus, C. D.; Scandiffio, M. I.; Rossell, C. E.; Maciel Filho, R.; Bonomi, A., Simulation of ethanol production from sugarcane in Brazil: economic study of an autonomous distillery. *Comput. Aided Chem. Eng.* **2010,** *28*, 733-738.
92. Unrean, P., Bioprocess modelling for the design and optimization of lignocellulosic biomass fermentation. *Bioresources and Bioprocessing* **2016,** *3* (1), 1.
93. Gernaey, K. V.; Lantz, A. E.; Tufvesson, P.; Woodley, J. M.; Sin, G., Application of mechanistic models to fermentation and biocatalysis for next-generation processes. *Trends Biotechnol.* **2010,** *28* (7), 346-354.
94. Van Impe, J. F., Power and limitations of model based bioprocess optimization. *Math. Comput. Simul.* **1996,** *42* (2-3), 159-169.
95. Link, H.; Christodoulou, D.; Sauer, U., Advancing metabolic models with kinetic information. *Curr. Opin. Biotechnol.* **2014,** *29*, 8-14.
96. Rosenbaum, M. A.; Henrich, A. W., Engineering microbial electrocatalysis for chemical and fuel production. *Curr. Opin. Biotechnol.* **2014,** *29,* 93-98.
97. Saha, R.; Chowdhury, A.; Maranas, C. D., Recent advances in the reconstruction of metabolic models and integration of omics data. *Curr. Opin. Biotechnol.* **2014,** *29*, 39-45.
98. Kim, J.; Reed, J. L., Refining metabolic models and accounting for regulatory effects. *Curr. Opin. Biotechnol.* **2014,** *29,* 34-38.
99. Villaverde, A. F.; Banga, J. R., Reverse engineering and identification in systems biology: strategies, perspectives and challenges. *J. Royal Soc. Interface* **2014,** *11* (91), 20130505.
100. Bailey, J. E., Mathematical modeling and analysis in biochemical engineering: past accomplishments and future opportunities. *Biotechnol. Progr.* **1998,** *14* (1), 8-20.
101. Ling, H.; Teo, W.; Chen, B.; Leong, S. S. J.; Chang, M. W., Microbial tolerance engineering toward biochemical production: from lignocellulose to products. *Curr. Opin. Biotechnol.* **2014,** *29*, 99-106.

102. Haus, S.; Jabbari, S.; Millat, T.; Janssen, H.; Fischer, R.-J.; Bahl, H.; King, J. R.; Wolkenhauer, O., A systems biology approach to investigate the effect of pH-induced gene regulation on solvent production by Clostridium acetobutylicum in continuous culture. *BMC Syst. Biol.* **2011,** *5* (1), 10.

103. Janssen, H.; Döring, C.; Ehrenreich, A.; Voigt, B.; Hecker, M.; Bahl, H.; Fischer, R.-J., A proteomic and transcriptional view of acidogenic and solventogenic steady-state cells of Clostridium acetobutylicum in a chemostat culture. *Appl. Microbiol. Biotechnol.* **2010**.

104. Grimmler, C.; Janssen, H.; Krausse, D.; Fischer, R. J.; Bahl, H.; Durre, P.; Liebl, W.; Ehrenreich, A., Genome-wide gene expression analysis of the switch between acidogenesis and solventogenesis in continuous cultures of Clostridium acetobutylicum. *J. Mol. Microbiol. Biotechnol.* **2011,** *20* (1), 1-15.

105. Shinto, H.; Tashiro, Y.; Kobayashi, G.; Sekiguchi, T.; Hanai, T.; Kuriya, Y.; Okamoto, M.; Sonomoto, K., Kinetic study of substrate dependency for higher butanol production in acetone–butanol–ethanol fermentation. *Process Biochem.* **2008,** *43* (12), 1452-1461.

106. Millat, T.; Janssen, H.; Bahl, H.; Fischer, R.-j.; Wolkenhauer, O. In *The pH-induced metabolic shift from acidogenesis to solventogenesis in Clostridium acetobutylicum—From experiments to models*, 5th Symposium on ESCEC, 2011.

107. Buehler, E. A.; Mesbah, A., Kinetic Study of Acetone-Butanol-Ethanol Fermentation in Continuous Culture. *PLoS ONE* **2016,** *11* (8), e0158243.

108. Shinto, H.; Tashiro, Y.; Yamashita, M.; Kobayashi, G.; Sekiguchi, T.; Hanai, T.; Kuriya, Y.; Okamoto, M.; Sonomoto, K., Kinetic modeling and sensitivity analysis of acetone–butanol–ethanol production. *J. Biotechnol.* **2007,** *131* (1), 45-56.

109. Liao, C.; Seo, S.-O.; Celik, V.; Liu, H.; Kong, W.; Wang, Y.; Blaschek, H.; Jin, Y.-S.; Lu, T., Integrated, systems metabolic picture of acetone-butanol-ethanol fermentation by Clostridium acetobutylicum. *Proceedings of the National Academy of Sciences* **2015,** *112* (27), 8505-8510.

110. Intellingen, SuperPro Designer Manual. In *New Jersey*, 2015.

111. Xue, C.; Zhao, J.; Chen, L.; Yang, S.-T.; Bai, F., Recent advances and state-of-the-art strategies in strain and process engineering for biobutanol production by Clostridium acetobutylicum. *Biotechnol. Adv.* **2017**.

112. Marquardt, W.; Harwardt, A.; Recker, S.; Viell, J.; Voll, A. In *11th International Symposium on Process Systems Engineering. Singapore*, Bio-based value chains of the future—an opportunity for process systems engineering, 2012; pp 19-28.

113. Majozi, T., Process Systems Engineering Approach to Synthesis of Batch Chemical Processes. In *Comput. Aided Chem. Eng.*, Karimi, I. A.; Srinivasan, R., Eds. Elsevier: 2012; Vol. 31, pp 83-90.

114. Motta, S.; Pappalardo, F., Mathematical modeling of biological systems. *Brief. Bioinform.* **2012,** *14* (4), 411-422.

115. Ponce, G. H.; Mirandaa, J. C.; Filhoa, R. M.; de Andradeb, R. R.; Wolfa, M. R., Simulation, Analysis and Optimization of Sugar Concentration in an In Situ Gas Stripping Fermentation Process for Bioethanol Production. *Chemical Engineering Transactions* **2014,** *37*, 295-300.

116. Sirigudi, R. R. Biomass to ethanol; Process simulation, validation and sensitivity analysis of a gasifier and a bioreactor. Oklahoma State University 2005.

117. Valderrama, J. O.; Toselli, L. A.; Faúndez, C. A., Advances on modeling and simulation of alcoholic distillation. Part 2: Process simulation. *Food and bioproducts processing* **2012,** *90* (4), 832-840.

118. Liu, J.; Wu, M.; Wang, M., Simulation of the Process for Producing Butanol from Corn Fermentation. *Industrial & Engineering Chemistry Research* **2009,** *48* (11), 5551-5557.

119. Nilsson, R.; Bauer, F.; Mesfun, S.; Hulteberg, C.; Lundgren, J.; Wännström, S.; Rova, U.; Berglund, K. A., Techno-economics of carbon preserving butanol production using a combined fermentative and catalytic approach. *Bioresour. Technol.* **2014,** *161* (0), 263-269.

120. Votruba, J.; Volesky, B.; Yerushalmi, L., Mathematical model of a batch acetone–butanol fermentation. *Biotechnol. Bioeng.* **1986,** *28* (2), 247-255.

121. Mayank, R.; Ranjan, A.; Moholkar, V. S., Mathematical models of ABE fermentation: review and analysis. *Crit. Rev. Biotechnol.* **2013,** *33* (4), 419-447.

122. AspenTech, A. T., Inc., , Aspen Plus V8.8 Help & Documentation. MA, USA, 2015.

123. Xue, C.; Zhao, J.; Liu, F.; Lu, C.; Yang, S.-T.; Bai, F.-W., Two-stage in situ gas stripping for enhanced butanol fermentation and energy-saving product recovery. *Bioresour. Technol.* **2013,** *135* (0), 396-402.

124. Lu, C.; Dong, J.; Yang, S.-T., Butanol production from wood pulping hydrolysate in an integrated fermentation–gas stripping process. *Bioresour. Technol.* **2013,** *143*, 467-475.

125. Cai, D.; Wang, Y.; Chen, C.; Qin, P.; Miao, Q.; Zhang, C.; Li, P.; Tan, T., Acetone–butanol–ethanol from sweet sorghum juice by an immobilized fermentation-gas stripping integration process. *Bioresour. Technol.* **2016,** *211*, 704-710.

126. Maddox, I. S.; Qureshi, N.; Roberts-Thomson, K., Production of acetone-butanol-ethanol from concentrated substrate using clostridium acetobutylicum in an integrated fermentation-product removal process. *Process Biochem.* **1995,** *30* (3), 209-215.

127. Ezeji, T. C.; Qureshi, N.; Blaschek, H. P., Production of acetone butanol (AB) from liquefied corn starch, a commercial substrate, using Clostridium beijerinckii coupled with product recovery by gas stripping. *J. Ind. Microbiol. Biotechnol.* **2007,** *34* (12), 771-777.

128. Qureshi, N.; Saha, B. C.; Hector, R. E.; Hughes, S. R.; Cotta, M. A., Butanol production from wheat straw by simultaneous saccharification and fermentation using Clostridium beijerinckii: Part I—Batch fermentation. *Biomass Bioenergy* **2008,** *32* (2), 168-175.

129. Ennis, B. M.; Marshall, C. T.; Maddox, I. S.; Paterson, A. H. J., Continuous product recovery by in-situ gas stripping/condensation during solvent production from whey permeate usingClostridium acetobutylicum. *Biotechnol. Lett* **1986,** *8* (10), 725-730.

130. Ezeji, T. C.; Karcher, P. M.; Qureshi, N.; Blaschek, H. P., Improving performance of a gas stripping-based recovery system to remove butanol from Clostridium beijerinckii fermentation. *Bioprocess Biosyst. Eng.* **2005,** *27* (3), 207-214.

131. Tashiro, Y.; Sonomoto, K., Advances in butanol production by clostridia. *Current Research, Technology and Education Topics in Applied Microbiology and Microbial Biotechnology (Microbiology Book Series, number# 2). FORMATEX, Spain* **2010**, 1383-1394.

132. Gapes, J. R., The economics of acetone-butanol fermentation: theoretical and market considerations. *J. Mol. Microbiol. Biotechnol.* **2000,** *2* (1), 27-32.

133. Sukumara, S.; Darkwah, K.; Seay, J., Process Simulations Supporting a Techno-Economic Framework to Optimize Biorefinery Supply Chains. *Comput. Aided Chem. Eng.* **2015,** *37*, 1985 - 1990.

134. Gençer, E.; Agrawal, R., Strategy to synthesize integrated solar energy coproduction processes with optimal process intensification. Case study: Efficient solar thermal hydrogen production. *Comput. Chem. Eng.* **2017**.

135. Yerushalmi, L. Physiological aspects of the acetone-butanol fermentation. Ph.D. Dissertation, McGill University, Montreal, Canada, 1985.

136. Sprenger, G. A., Carbohydrate metabolism in Zymomonas mobilis: a catabolic highway with some scenic routes. *FEMS Microbiol. Lett.* **1996,** *145* (3), 301-307.

137. Leksawasdi, N.; Joachimsthal, E. L.; Rogers, P. L., Mathematical modelling of ethanol production from glucose/xylose mixtures by recombinant Zymomonas mobilis. *Biotechnol. Lett* **2001,** *23* (13), 1087-1093.

138. Thibault, J., Advances in Process Systems Engineering: Multi-objective optimization: techniques and applications in chemical engineering. In *Net flow and rough sets: two methods for ranking the Pareto domain*, First ed.; World Scientific Singapore: 2009.

139. Reeves, C. R., A genetic algorithm for flowshop sequencing. *Computers & Operations Research* **1995,** *22* (1), 5-13.

140. Kang, Q.; Appels, L.; Tan, T.; Dewil, R., Bioethanol from lignocellulosic biomass: current findings determine research priorities. *The Scientific World Journal* **2014,** *2014*.

141. Xue, C.; Zhao, J.; Lu, C.; Yang, S. T.; Bai, F.; Tang, I., High-titer n-butanol production by clostridium acetobutylicum JB200 in fed-batch fermentation with intermittent gas stripping. *Biotechnol. Bioeng.* **2012,** *109* (11), 2746-2756.

Vita

Kwabena Darkwah

Place of Birth: Accra, Ghana

Educational institutions attended and degrees already awarded:

Master of Science in Chemical Engineering**,** North Carolina A&T State University, Greensboro, NC, 2013.

Bachelor of Science in Chemical Engineering, Kwame Nkrumah University of Science and Technology, Ghana, 2006.

Professional Positions held:

Graduate Research Assistant, Chemical and Material Engineering Department, University of Kentucky, August 2013 – February 2018.

Teaching Assistant, Chemical and Material Engineering Department, University of Kentucky, August 2013 – August 2017.

Graduate Research Assistant, Chemical, Biological and Bioengineering Department, North Carolina A&T State University, August 2010 – May 2010.

Quality Assurance Manager, Commodities Processing Industries, Ltd., Tema, Ghana, 2009 – 2010.

Production Supervisor, Commodities Processing Industries, Ltd., Tema, Ghana, 2008 – 2009.

Plant and Process Engineer, Rex Laboratories Ltd., Weija, Ghana, 2006 – 2008.

Trainee Engineer, Tema Oil Refinery Ltd., Tema, Ghana, 2006 – 2007.

Scholastic and Professional Honors:

First place (out of 20), 6th International Forum on Sustainable Manufacturing, Institute for Sustainable Manufacturing, Lexington, Kentucky, December 2017

Honorary Mention, Sustainability and Sustainable Biorefineries Poster Session, AICHE Annual Meeting, San Francisco, California, November 2016.

Second Place (out of 20) in Poster Competition, Materials and Chemical Engineering Graduate Students Association Spring Symposium, University of Kentucky, May 2016.

Recipient of the USEC Inc. Graduate Fellowship, University of Kentucky College of Engineering, February 2016.

Best Presentation, out of 6 in session 192 c (Biofuels Production: Design, Simulation, and Economic Analysis), Annual AICHE Meeting in Salt Lake City, November, 2015.

Second place (out of 47), Sustainability and Sustainable Biorefineries Poster Session, AICHE Annual Meeting, Salt Lake City, Utah, November 2015.

Second place (out of 20), 5th International Forum on Sustainable Manufacturing, Institute for Sustainable Manufacturing, Lexington, Kentucky, September 2015

Certificate of Scholarly Accomplishment and Excellence in Academic Performance, International Students & Scholars Office, North Carolina A & T State University, 2012

Certificate of Scholarly Accomplishment and Excellence in Academic Performance International Students & Scholars Office, North Carolina A & T State University, 2011.

Professional Publications:

Darkwah, K., Knutson, L. B, Seay, R. J. “A Perspective on Challenges and Prospects for Applying Process Systems Engineering Tools to Fermentation-Based Biorefineries.” *ACS Sustainable Chemistry & Engineering.* DOI: 10.1021/acssuschemeng.7b03762

Darkwah, K., L. Wang, and A. Shahbazi. "Simultaneous saccharification and fermentation of sweet sorghum after acid pretreatment." *Energy Sources, Part A: Recovery, Utilization, and Environmental Effects* 38.10 (2016): 1485-1492.

Sukumara, S., **Darkwah, K.**, Seay, R. J. “Process Simulations Supporting a Techno-Economic Framework to Optimize Biorefinery Supply Chains.” *12th International Symposium on Process Systems Engineering and 25th European Symposium on Computer Aided Process Engineering*. 37 (2015): 1985-1990 (Conference Paper).

Darkwah, K., Knutson, L. B, Seay, R. J. "Multi-objective versus single-objective optimization of batch bioethanol production based on a time-dependent fermentation model." In preparation to be submitted to *Clean Technologies and Environmental Policy Journal*.

Darkwah, K., Seay, R. J., Knutson, L. B. "Aspen Plus unsteady state fed-batch fermentation and *in situ* gas stripping simulations and analyses." In preparation for submission to *Bioprocess & Biosystems Engineering Journal* for publication.

Darkwah, K., Seay, R. J., Knutson, L. B. "*In silico* analyses of unsteady state batch Acetone-Butanol-Ethanol (ABE) fermentation and *in situ* gas stripping." In preparation for submission to *Bioprocess & Biosystems Engineering Journal* for publication.

Kwabena Darkwah

February 9, 2018