2023

State Estimation, Covariance Estimation, and Economic Optimization of Semi-Batch Bioprocesses

Ronald Hunter Alexander
rhalexander@mix.wvu.edu

State Estimation, Covariance Estimation, and Economic Optimization of Semi-Batch Bioprocesses

Ronald H. Alexander

Dissertation submitted to the
Benjamin M. Statler College of Engineering and Mineral Resources
At West Virginia University

In partial fulfillment of the requirements for the degree of

Doctor of Philosophy in
Chemical Engineering

Dr. Fernando V. Lima, Ph.D., Chair
Dr. Debangsu Bhattacharyya, Ph.D.
Dr. Stephen Zitney, Ph.D.
Dr. Mario Perhinschi, Ph.D.
Dr. Marcelo P. A. Ribeiro, Ph.D.
Dr. Paolo Pezzini, Ph.D.

Department of Chemical and Biomedical Engineering

Morgantown, West Virginia
2023

Keywords: Nonlinear State Estimation, Covariance Estimation, Autocovariance Least-Squares, Semi-Batch Optimization, Process Economics

Abstract

State Estimation, Covariance Estimation, and Economic Optimization of Semi-Batch Bioprocesses

Ronald H. Alexander

One of the most critical aspects of any chemical process engineer is the ability to gather, analyze, and trust incoming process data as it is often required in control and process monitoring applications. In real processes, online data can be unreliable due to factors such as poor tuning, calibration drift, or mechanical drift. Outside of these sources of noise, it may not be economically viable to directly measure all process states of interest (e.g., component concentrations). While process models can help validate incoming process data, models are often subject to plant-model mismatches, unmodeled disturbances, or lack enough detail to track all process states (e.g., dissolved oxygen in a bioprocess). As a result, directly utilizing the process data or the process model exclusively in these applications is often not possible or simply results in suboptimal performance.

To address these challenges and achieve a higher level of confidence in the process states, estimation theory is used to blend online measurements and process models together to derive a series of state estimates. By utilizing both sources, it is possible to filter out the noise and derive a state estimate close to the true process conditions. This work deviates from the traditional state estimation field that mostly addresses continuous processes and examines how techniques such as extended Kalman Filter (EKF) and moving horizon estimation (MHE) can be applied to semi-batch processes. Additionally, this work considers how plant-model mismatches can be overcome through parameter-based estimation algorithms such as Dual EKF and a novel parameter-MHE (P-MHE) algorithm. A galacto-oligosaccharide (GOS) process is selected as the motivating example as some process states are unable to be independently measured online and require state estimation to be implemented. Moreover, this process is representative of the broader bioprocess field as it is subject to high amounts of noise, less rigorous models, and is traditionally operated using batch/semi-batch reactors.

In conjunction with employing estimation approaches, this work also explores how to effectively tune these algorithms. The estimation algorithms selected in this work require careful tuning of the model and measurement covariance matrices to balance the uncertainties between the process models and the incoming measurements. Traditionally, this is done via ad-hoc manual tuning from process control engineers. This work modifies and employs techniques such as direct optimization (DO) and autocovariance least-squares (ALS) to accurately estimate the covariance values. Poor approximation of the covariances often results in poor estimation of the states or drives the estimation algorithm to failure.

Finally, this work develops a semi-batch specific dynamic real-time optimization (DRTO) algorithm and poses a novel costing methodology for this specific type of problem. As part of this costing methodology, an enzyme specific cost scaling correlation is proposed to provide a realistic approximation of these costs in industrial contexts. This semi-batch DRTO is combined with the GOS process to provide an economic analysis using *Kluyveromyces lactis* (*K. lactis*) β-galactosidase enzyme. An extensive literature review is carried out to support the conclusions of the economic analysis and motivate application to other bioprocesses.

Dedicated to

My mom Annmarie Alexander

Acknowledgments

I would like to thank everyone who I have worked with during my time as both a BS and PhD student at WVU. The last 8 years of my life at WVU have been a transformational experience and it is incredibly hard to believe how fast it all went by!

First, I would like to specifically thank Dr. Fernando V. Lima for kickstarting my interest in the field of process systems engineering (PSE) research and allowing me to join the CODES research group during my junior year of undergrad. It is hard to state how significant of an impact this initial opportunity has had on my life, both professionally and personally. Professionally, this initial research opportunity led to Dr. Lima becoming my Summer Undergraduate Research Experience (SURE) advisor between my Junior and Senior years. This experience ultimately led to me choosing to pursue my PhD at WVU under the advisement of Dr. Lima. I knew I wanted to continue working in the CODES group and as a result only applied to WVU for graduate school with the intention of remaining in the CODES group. Without this mentoring and support during my undergraduate and graduate years, I would not be where I am today.

I am also extremely grateful for the financial support I have received from Statler College in the form of a Statler Fellowship. This funding source gave me the opportunity and flexibility to explore the PSE topics discussed in this dissertation with few restrictions. As a result, I would like to especially thank Benjamin Statler for establishing this fellowship. Additionally, I would also like to thank both the Department of Chemical and Biomedical Engineering and Statler College for additional financial support.

Of course, I would also like to thank each one of my committee members: Dr. Debangsu Bhattacharyya, Dr. Stephen Zitney, Dr. Mario Perhinschi, Dr. Marcelo P. A. Ribeiro, and Dr. Paolo Pezzini for their time, support, and suggestions!

In terms of former CODES members, I would like to give special thanks to Dr. Rebecca Kim and Dr. San Dinh, as both were extremely helpful mentors during both my time as a BS and PhD student in the CODES group. Additionally, I would also like to thank Dr. Brent Bishop, Hunter Barber, Lillian Bischof, Ashley McCullough, Selorme Agbleze, Victor Alves, Daniel Kestering, Vitor Gama, Claudemi Nascimento, Beatriz Dantas, Bernardo Vecchio, Savannah Sakhai, Krishna Busam, Davi Oliveira, Michael Fouts, and Antonio Mascaro.

Finally, I would also like to thank Rebekah Rice and my mom Annmarie Alexander for their support and always being there for me!

Thank you very much!

Montani Semper Liberi

Table of Contents

List of Tables

List of Figures

Nomenclature

Notation

a	Intercept P-quantity Q correlation
$\bar{A}_k$	Linearized system matrix
b	P-quantity Q correlation slope
b_{avg}	Average P-quantity Q correlation slope
b_{enz}	Enzyme specific P-quantity Q correlation slope
B	Input matrix
$\bar{C}_k$	Linearized controllability matrix
C_{enz}	Enzyme feed concentration [M]
C_{lac}	Lactose feed concentration [M]
Di	Total disaccharide concentration [M]
e	Free enzyme concentration
E	Active enzyme concentration [M]
$f(\hat{x}_k, u_k)$	Nonlinear reactor model
G	System noise matrix
$\bar{G}_k$	Linearized system noise matrix
Gal	Galactose concentration [M]

Glb	Galactobiose concentration [M]
Glu	Glucose concentration [M]
GOS3	Total trisaccharide concentration [M]
GOS4	Total tetrasaccharide concentration [M]
$h(\hat{x}_k^-)$	Nonlinear system function
$init_{lac}$	Initial lactose
J	Number of data sets
k	Discrete time index
L_k	Kalman filter gain
Lac	Lactose concentration [M]
MW	Molecular weight
N	Horizon length
N_{ALS}	Number of lags
P	Unit price of chemical [$/kg]
P_0^-	Initial forecasted estimator covariance matrix
P_B	Bulk unit price [$/kg]
P_{k+1}^-	Forecasted estimator covariance matrix
P_k	Estimator covariance matrix
P_1	Single unit price [$/kg]

Q	Model covariance matrix
Q_B	Representative bulk amount [g]
Q_{corr}	Purchase quantity [g]
Q_1	Single unit purchase quantity [g]
R	Measurement covariance matrix
$\mathcal{R}_k$	Autocovariance matrix
t_{batch}	Batch time [min]
t_{prep}	Preparation time [min]
Tet	Glucose rich tetrasaccharide concentration [M]
Tetg	Purely galactose rich tetrasaccharide concentration [M]
Tri	Glucose rich trisaccharide concentration [M]
Trig	Purely galactose rich trisaccharide concentration [M]
u	Control moves
u_{enz}	Enzyme feed flowrate [m^3/min]
u_{lac}	Lactose feed flowrate [m^3/min]
$\hat{V}_T^-(x_{T-N})$	Arrival cost
v	Measurement noise
V	Volume [m^3]
V_{max}	Maximum reactor volume [m^3]

w	Model noise
$\bar{x}_0$	Initial guess of the system states
$\hat{x}_0^-$	Initial forecasted state estimate
$\hat{x}_{k+1}^-$	Forecasted state estimate
$\hat{x}_k$	State estimate
x_{model}	Model output
y_k	Measurement
$\mathbb{y}_k$	Innovation sequence
z	Augmented state vector
$\oplus$	Direct sum
$\otimes$	Kronecker product
$\|X\|_Y^2$	Quadratic Error

Acronyms

ACM	Autocovariance matrix
ALS	Autocovariance least-squares
A. oryzae	*Aspergillus oryzae*
CSTR	Continuous stirred-tank reactor

DCS	Distributed control system
DO	Direct optimization
DRTO	Dynamic real-time optimization
EKF	Extended Kalman filter
GOS	Galacto-oligosaccharide
HPLC	High performance liquid chromatography
KF	Kalman filter
K. lactis	*Kluyveromyces lactis*
LTI	Linear time-invariant
MHE	Moving horizon estimation
MPC	Model predictive control
MV	Manipulated variable
NEGA	Noise covariance matrices estimation with Gaussianity assessment
OPC	Open Platform Communication
P-MHE	Parameter-based moving horizon estimation
PID	Proportional-integral-derivative
PSE	Process systems engineering
PSO	Particle swarm optimization

RTO	Real-time optimization
SE	Square Error
SP	Setpoint
SQP	Sequential quadratic programming
UKF	Unscented Kalman filter

Greek Letters

Δt	Discretization time
ε_k	State estimation error
$\zeta_{p,N}$	Permutation matrix
θ	Kinetic parameter covariance matrix
ρ	Solution density
ρ_{kin}	Kinetic parameters
Ψ	Augmented model covariance matrix

State Estimation, Covariance Estimation, and Economic Optimization of Semi-Batch Bioprocesses

Chapter 1 Introduction

One of the most critical aspects of any chemical process is the ability to gather, analyze, and trust incoming process data. Real processes use a variety of online sensors and offline measurements derived from process samples, correlations, process models, and even operator knowledge as sources of information to help informed decision making at various levels. Although all data sources have some inherent value and potential use, not all data are equally valuable for making decisions for online monitoring and control of chemical plants. For example, operators may provide useful insight regarding the mechanical health of process equipment (e.g., degradation or leaks) but are unable to manually gather temperature and pressure readings to operate a plant in an online manner.

As a result, all modern chemical plants heavily rely on online sensors such as thermocouples, pressure gauges, inline gas chromatography units, and flowmeters to provide a constant stream of information to the distributed control system (DCS). In particular, the DCS is the brain of the modern chemical plant and is responsible for interpreting the incoming process data and responding with the appropriate control action (Segovia & Theorin, 2013). Operators and control engineers can override the DCS, but much of the time the DCS is at the helm of monitoring and controlling the plant. Although variations in DCS vendor can influence the inclusion of specific features such as nonlinear control, Industry 4.0 integration, or integration with external software tools, nearly all commercial DCS implementations will directly supervise an ongoing process.

Most commonly, this supervision is performed using proportional-integral-derivative (PID) control algorithms directly embedded into the DCS software, where the incoming process data is used to determine the current process states. If a process state or output does not match its specified setpoint (SP) within the PID, corrective control action will be taken by varying a manipulated variable (MV) (Rice, 2010). It should be noted that model predictive control (MPC) is becoming more common across DCS vendors, but this does not modify the DCS's responsibility for regulating the process through variation of the MVs (Rockwell Automation, 2016). Figure 1.1 shows how a basic PID control loop interacts with the DCS.

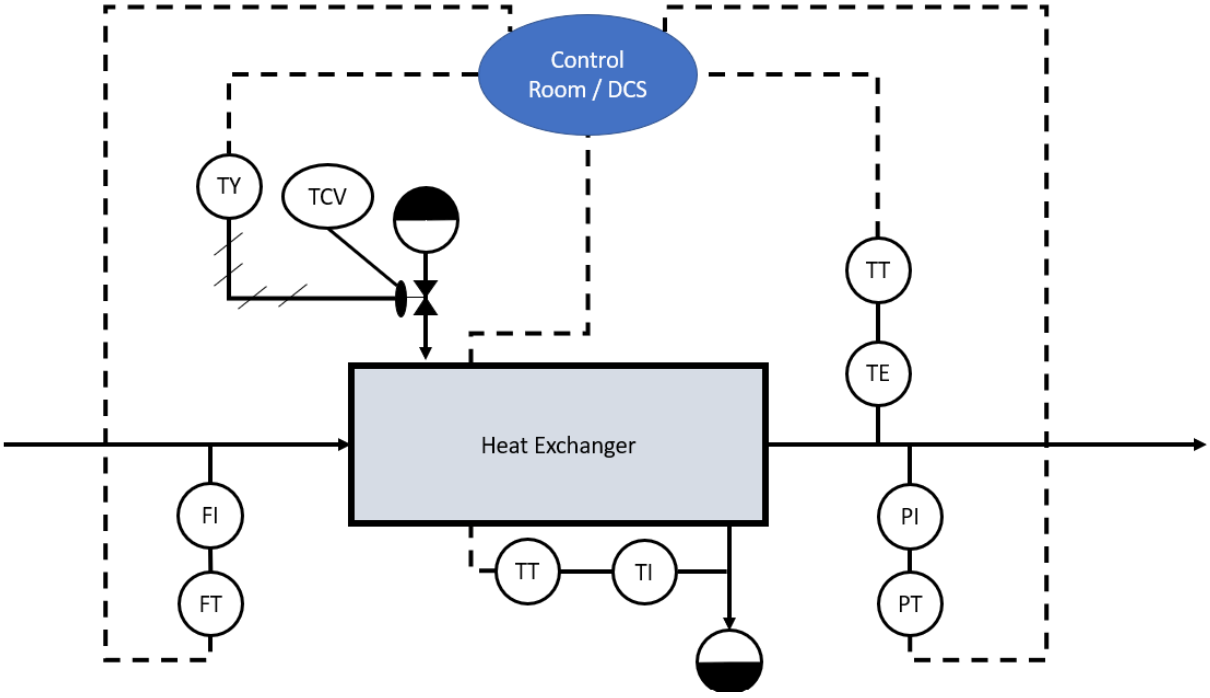

Figure 1.1 Basic PID loop interaction with DCS

This example control loop is designed to regulate the effluent temperature of the process stream leaving the heat exchanger by varying the flowrate of the utility stream.

For this example, the temperature of the effluent stream is measured using an online temperature sensor (TE) which feeds this information to the DCS inside the control room using a transmitter (TT). Tuned PID controllers inside the DCS software read this signal and update the flowrate of the utility steam by relaying (TY) an updated valve stem position for the utility steam control valve (TCV), if the effluent stream temperature does not match the specified setpoint. Even in this basic control loop, additional process states such as pressure (PI) or flow rate (FI) may be tracked and transmitted (PT and FT, respectively) to the DCS to help validate temperature measurements or to track difficult to measure process phenomena such as fouling.

In theory, raw online process data is all that is required to conduct feedback control of a process. However, in real processes, data from online sensors may be unreliable due to measurement errors, poor tuning, calibration drifts or mechanical failures. For some processes, there are no online sensors that can measure key process states of interest, or in many cases the measurements cannot be collected in an economically efficient manner. State estimation techniques can be applied to reduce these noise effects and provide a higher amount of confidence in the process data. In the DCS example shown above, a state estimation algorithm could blend a model prediction for the outlet temperature with the available measurements to filter out noise and provide more accurate states for feedback control.

Most of the literature involving state estimation is devoted to its application to continuous processes that are operating at steady-state or those which are undergoing dynamic transitions from one steady-state to another. This trend is largely driven by the popularity and success of continuous processes in the petrochemical and commodity

chemical industries, power plants and other large-scale highly automated processes with high production rates (Chen, 2017). Due to this emphasis on continuous processes, entire segments of the chemical process industry that rely heavily on batch processes, such as the pharmaceutical and food industries, are not addressed thoroughly in the estimation literature.

State estimation could be particularly attractive to the pharmaceutical industry as it is beholden to strict quality control regulations. Although some attempts are being made to convert this industry to continuous operation, there are still many challenges that must be addressed such as, economic viability of switching, technology not currently existing (in particular for biological products), or process safety (Chen, 2017; Lee, 2017). Likely, batch processes will remain common place for certain aspects of the pharmaceutical industry as well as food processing industry. As a result, there is a clear motivation to develop batch or semi-batch specific state estimation algorithms to address this gap in the literature.

The development of semi-batch specific state estimation algorithms is not enough however, as these algorithms require careful tuning to correctly filter out the noise and derive accurate estimates of the process states (Alexander et al., 2020). Most commonly process control engineers will attempt to quantify the uncertainty of various data sources to correctly weigh the significance of each piece of information. For example, in Figure 1.1 if a model prediction is available and accurate, then the model will have a lower uncertainty and will be relied upon more than the online measurement in the state estimation algorithm. Over time, if the model becomes a poor representation of the plant due to unmodeled disturbances (e.g., fouling), then the uncertainty of the model may

become large enough such that the measurement would be relied upon more. Even in this simple example, identification and quantification of the model and measurement uncertainties is a non-trivial task. This motivates the development of specific techniques to derive the covariances that characterize these uncertainty sources to help achieve more accurate state estimation values for industrial processes. This is particularly critical in the food and pharmaceutical industries where accurate state estimates could be used to help ensure that quality control standards are reached.

Although not directly related to online control, some DCS implementations go well beyond process control tasks and are beginning to adapt to the new paradigm of Industry 4.0 combined with internet of things (IoT) (Badii et al., 2020). The increased access to plant data, ability to share data through communication protocols such as Open Platform Communication (OPC), and expansion in computing power has revolutionized scheduling, optimization, supply chain management, and tracking of relevant performance indicators (Bellini et al., 2022). As a result of this data access and connectivity, there are many possible ways of improving traditional control approaches. For example, economic-based dynamic real-time optimization (DRTO) algorithms written in an external coding language such as Python or MATLAB could receive updated economic data regarding product value, feedstock cost, or other operating expenses from supply chain experts, and then update setpoints within the plant. Such automated algorithms would be extremely useful for batch or semi-batch processes which are traditionally controlled using repeatable recipes, programmable logic controllers (PLC), or other nonoptimal forms of control. Once again, this could be useful in the pharmaceutical or industrial food industry as market demands and feedstock costs are

constantly varying, and few tools are available to rigorously optimize and optimally control these processes due to the batch and semi-batch nature of these industries.

Overall, there are many opportunities to develop and apply process systems engineering (PSE) tools to achieve better process monitoring and state estimation for improved bioprocess economics. In particular, the emphasis on continuous processes has created a unique gap in the literature where batch and semi-batch specific PSE tools can be developed. As a result, this motivates the 4 specific aims of this dissertation listed below:

1) **Develop semi-batch specific state estimation algorithms**. Various classes of recursive and optimization-based algorithms will be developed and implemented for semi-batch specific process applications. This will include the development of a new parameter-based moving horizon estimator (P-MHE). As part of this work, a discussion on estimator robustness and computational cost is included to facilitate further application of these tools.
2) **Employ systematic tools for deriving estimates of model and measurement covariances**. To improve upon the standard ad-hoc covariance estimation techniques, systematic tools will be developed, tested, and applied to the algorithms in Aim 1. Specifically, direct optimization (DO) and autocovariance least-squares (ALS) techniques will be further developed. These systematic algorithms will yield consistent and accurate approximations of the system covariances, thus leading to improved estimation accuracy over ad-hoc tuning methods.

3) **Propose a novel costing approach as part of a semi-batch specific DRTO framework**. To address the semi-batch DRTO gap in the literature, an entirely novel semi-batch specific costing methodology and DRTO problem formulation are proposed. The novel costing approach serves as the objective function for the DRTO and allows the optimal feeding policy to be derived for maximization of the process economics considering factors such as batch time, yield, and product quality. As a part of this aim, an enzyme specific cost scaling approach is developed which allows bulk industrial cost to be estimated from laboratory-scale pricing data.
4) **Apply state estimation, covariance estimation, and semi-batch DRTO tools to a bioprocess**. To test the approaches developed in the individual aims, a semi-batch galacto-oligosaccharide (GOS) process is used as the case study. This process requires state estimation as not all process states are directly measurable. Furthermore, there are few economic studies of this process (particularly in semi-batch operation), so a robust-economic based DRTO study is needed to determine its optimal feeding policy.

1.1 Research Outputs

The specific contributions of this work are: i) development of a new joint parameter and state estimation algorithm known as P-MHE. This algorithm is designed to address plant model mismatches common in batch processes and modifies existing approaches to allow for direct estimation of the kinetic parameters and process states concurrently with a much lower computational time than traditional approaches; ii) propose and test

alternative DO techniques to improve this covariance estimation method; iii) development of a MATLAB specific version of the ALS technique for semi-batch processes with application to the GOS process; iv) creation of a novel semi-batch specific economic DRTO approach that is generalizable to other semi-batch process; and v) propose an enzyme specific cost scaling approach to estimate bulk enzyme costs for economic studies associated with the GOS process.

Overall, these contributions have resulted or will result in the following publications:

Journal Publications

1. Alexander, R., Campani, G., Dinh, S., & Lima, F. V. (2020). Challenges and Opportunities on Nonlinear State Estimation of Chemical and Biochemical Processes. *Processes*, *8*(11), 1462. https://doi.org/10.3390/pr8111462
2. Schultz, G., Alexander, R., Lima, F. V., Giordano, R. C., & Ribeiro, M. P. A. (2021). Kinetic modeling of the enzymatic synthesis of galacto-oligosaccharides: Describing galactobiose formation. *Food and Bioproducts Processing*, *127*, 1-13. https://doi.org/10.1016/j.fbp.2021.02.004
3. Alexander, R., Dinh, S., Schultz, G., Ribeiro, M. P. A., & Lima, F. V. (2023). State and covariance estimation of a semi-batch reactor for bioprocess applications. *Computers & Chemical Engineering*, *172*, 108180. https://doi.org/10.1016/j.compchemeng.2023.108180
4. Alexander, R., Maione, N. R., Ribeiro, M. P. A, & Lima, F. V. (In preparation) Economic Analysis of Galacto-oligosaccharide Production using a Semi-batch Dynamic Real-Time Optimization Framework.

5. Alexander, R., & Lima, F. V. (In Preparation) Development of an Autocovariance Least-Squares Covariance Estimation Technique for Semi-Batch Bioprocesses.

Selected Conference Presentations

1. Alexander, R., Schultz, G., Ribeiro, M. P. A. & Lima, F. V. (2020) Modeling and Nonlinear State Estimation for Advanced Process Control of the Enzymatic Conversion of Lactose into Value-Added Products. Oral presentation at AIChE Annual Meeting (Virtual)
2. Alexander, R., Dinh, S., Schultz, G., Ribeiro, M. P. A. & Lima, F. V. (2021) Keynote: Multi-Objective Optimization, State Estimation, and Advanced Control of a Semi-Batch Process for the Enzymatic Conversion of Lactose into Value-Added Products. Oral presentation at AIChE Annual Meeting in Boston, MA
3. Alexander, R., Ribeiro, M. P. A., & Lima, F. V. (2022) An Integrated State Estimation, Covariance Estimation, and Optimal Control Framework of a Semi-Batch Reactor for Bioprocess Applications. Oral presentation at AIChE Annual Meeting in Phoenix, AZ
4. McCullough, A., Bischof, L., Alexander, R., Agbleze, S., & Lima, F. V. (2023) Development of an Algorithm to Evaluate the Performance and Economic Feasibility of Expansion of Solar and Wind Power Generation in the Appalachian Region. Oral presentation at Clearwater Clean Energy Conference (Hybrid)

1.2 Dissertation Organization

To facilitate discussing the first 3 aims of this dissertation, each aim is broken down into its own independent chapter. Chapter 2 summarizes the current literature of each aim and helps to provide context for each topic. Chapter 3 is dedicated to developing, implementing, and testing various estimation algorithms on semi-batch bioprocesses. This chapter also introduces the GOS process that is used as the case study system for much of this dissertation. Chapter 4 builds upon Chapter 3, by discussing how ALS can be used to derive accurate estimates of the model and measurement covariance matrices while maintaining low computational time. Chapter 5 focuses on developing a novel semi-batch specific costing approach with an emphasis on bioprocesses. Additionally, this chapter also presents a novel DRTO algorithm for optimizing semi-batch processes with respect to process economics. Finally, Chapters 6 and 7 document the conclusions of this dissertation and future recommendations that can be used to further develop and improve upon the work presented.

Chapter 2 Literature Review

The fields of state estimation, covariance estimation, and dynamic optimization considering process economics are distinct areas of research that can be difficult to connect. Much of the published literature in these fields focuses on a single topic, thus there are limited publications discussing how these fields can synergistically work together. As a result, this literature review will be broken down into distinct sections that will focus on a single topic independently. Some of the information from this section was previously discussed and published in a state estimation review paper. Specifically, segments of Sections 2.1 and 2.2 are derived from Alexander et al. (2020) which can be referenced for a broader literature review of the estimation field. The content presented in this chapter is intended to provide enough context for the proceeding chapters of this dissertation.

2.1 State Estimation

Simply stated, state estimation is a mathematical approach for improving knowledge of process states by blending measurements and process models together. This is essential as both measurements and process models are imperfect and thus can provide unrealistic information regarding where the true process state values are. Measurements provide real-time information about the process states but are frequently subject to noise, poor calibration, instrumentation failure, or maybe unavailable for certain states (e.g., concentration, density). Additionally, measurement devices are unable to predict the future behavior of processes which may be required to avoid unsafe operating regions or to reach production targets (Alexander et al., 2020).

Process models can provide predictive behavior but are frequently developed offline and tuned using historical data sets that may not reflect the current conditions of the plant. For example, a kinetic model may be developed for a packed bed reactor (PBR) using historical data that does not account for coking or gradual deactivation of the catalyst experienced in the plant. Over time, the state values predicted by the model will deviate further and further from the true state values seen in the plant. By combining these two data sources, it is possible to leverage the benefits of both process models and measurements to gain a better approximation of the true process states.

Historically, many state estimation techniques can trace their origins back to the Bayesian rule developed by Thomas Bayes in the 18th century (Bayes, 1763). Modern algorithms can more directly trace their roots to Rudolf Kalman and the development of the Kalman filter (KF) in the 1960s (Kalman, 1960). Canonically, the KF is a recursive linear filter which employs a linear state-space model of the process with measurements to derive estimates of the process states using the 2-step recursive procedure shown in Figure 2.1.

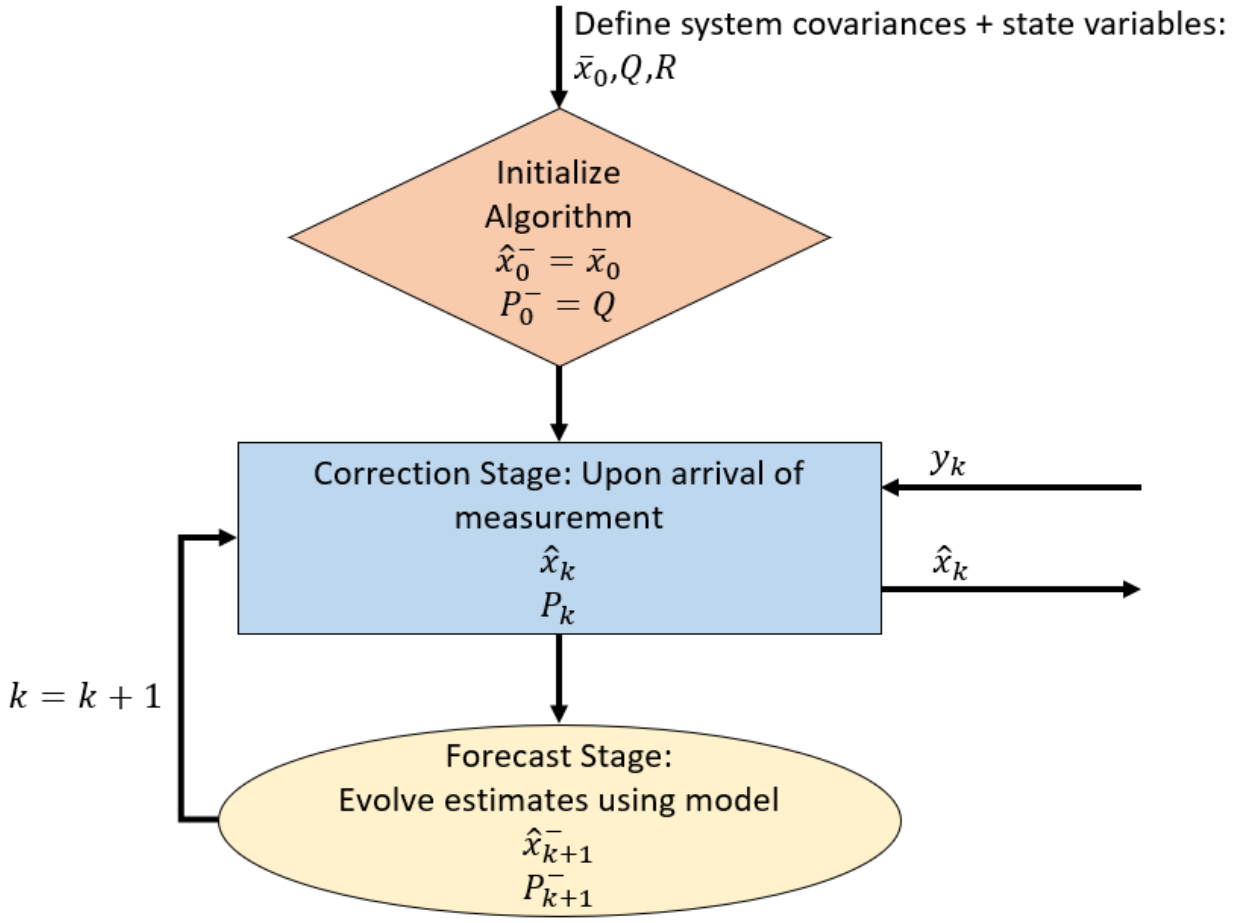

Figure 2.1 Recursive estimation algorithm procedure

Figure 2.1 shows the basic structure of most recursive estimation techniques and begins by providing an initial guess of the system states ($\bar{x}_0$), an initial guess of the measurement covariance matrix (R) and an initial guess of the model covariance matrix (Q) to derive the initial forecasted state estimate ($\hat{x}_0^-$) and the initial forecasted estimator covariance matrix (P_0^-). Upon collection of the first measurement (y_k), the correction step of the algorithm is initialized to derive the current state estimates ($\hat{x}_k$) and the estimator covariance matrix (P_k). Once this step is completed, the process model is used to conduct the forecast step of the algorithm to derive the new set of forecasted state

estimates ($\hat{x}_{k+1}^{-}$) and forecasted estimator covariance matrix (P_{k+1}^{-}). Finally, the estimation loop is closed by iterating the discrete time index (k) ahead one step and reconducting the correction step upon collection of a new measurement. This 2-step recursive process continues until the algorithm is terminated by the user.

To address nonlinearities found in many processes (particularly chemical processes), techniques such as model linearization and nonlinear transformations were introduced to develop the extended Kalman filter (EKF) (Rawlings et al., 2022) and unscented Kalman filter (UKF) (Julier & Uhlmann, 2004). From here, other forms of EKF and UKF have been developed to address differential algebraic equations (DAE) (Purohit & Patwardhan, 2018), parameter uncertainty (Sun et al., 2008), and other unique situations.

Although these recursive techniques are computationally efficient, there are several drawbacks to their industrial use, with the most significant being the inability to guarantee feasibility. EKF has many well documented examples of failing in the literature and cannot easily be modified to prevent this from occurring (Wilson et al., 1998; Haseltine & Rawlings, 2005; Alexander et al., 2020). There have been some ad-hoc fixes for EKF such as “clipping” the state estimates to feasible values if they become infeasible (Kolås et al., 2009), but this presents its own set of issues and generally produces poor estimation results. Due to this unconstrained nature, a variety of factors can lead to EKF instability (Ungarala et al., 2007) including multiple states satisfying the steady-state measurement (i.e., multiple solutions), poor initial guess of the system states (Haseltine & Rawlings, 2005), poor linearization, incorrect covariances, and even limited arithmetic precision leading to round-off operations (Verhaegen & Van Dooren, 1986; Lu et al.,

2007). For certain nonlinear systems, first-order linearization can cause large errors for the mean and covariance calculations or they may simply be nondifferentiable, thus not allowing the Jacobian matrices to be calculated (Zarei & Shokri, 2014). A very commonly cited paper examining EKF failure (and the benefits of alternative methods) is Haseltine & Rawlings (2005) which ultimately led to a rigorous discussion of this topic throughout the PSE community.

A commonly cited response paper is from Schneider & Georgakis (2013) which argues that under appropriate initialization conditions with effective design algorithms, EKF is an attractive state estimation method. Specifically, it is argued that if the initial covariance guess is specified accordingly (Valappil & Georgakis, 2000), the EKF can converge (Schneider & Georgakis, 2013). In terms of remediating numerical issues, square-root filtering methods have been proposed (Bellantoni & Dodge, 1967; Lu et al., 1992). Outside of these methods, numerical stability has been improved by replacing conventional matrix inversion methods with the Moore-Penrose matrix pseudoinversion (Kulikov & Kulikova, 2019). Additional publications in the literature have developed alternative filter formulations including the constrained extended Kalman filter (CEKF) (Ungarala et al., 2007), the constrained cubature Kalman filter (CCKF) (Zarei & Shokri, 2014), and other Kalman filter formulations that allow equality and inequality constraints to be incorporated. Overall, the underlying unconstrained nature of EKF requires applying a variety of unique filter modifications to promote feasible and derive reliable state estimation results.

In industrial applications, the infeasibility of traditional EKF can cause systematic control problems if the resulting state estimates from the filter are sent to the control

algorithms inside the DCS. Although alternative EKF formulations could be applied to remediate some of these infeasibilities, these algorithms have not been widely deployed in industrial applications and would require the control engineer to identify the source of the Kalman filter failure and choose an appropriate algorithm to remediate that challenge. Overall, this is more of an academic approach to applying state estimation as the solution is not scalable or immediately robust across all potential applications. For cases where feasibility is a concern, an alternative class of optimization-based estimation algorithms is available that can immediately be implemented into a wide variety of processes.

Many optimization-based state estimation approaches structure the objective function as a least-squares problem that incorporates data from the process model and measurements, along with their model and measurement covariances, respectively. The most common algorithm from this class is the moving horizon estimation (MHE) which uses a sliding window of recent measurements (Rawlings et al., 2022). The sliding window limits the computational cost of the optimization problem by reducing the estimation to recent data points instead of the full information problem (i.e., deriving estimates for the entire process). As MHE uses optimization, it is straightforward to embed a variety of constraints such as non-negative values, min/max value, nonlinear constraints, mass balances, and energy balances. The moving horizon approach can be seen below in Figure 2.2

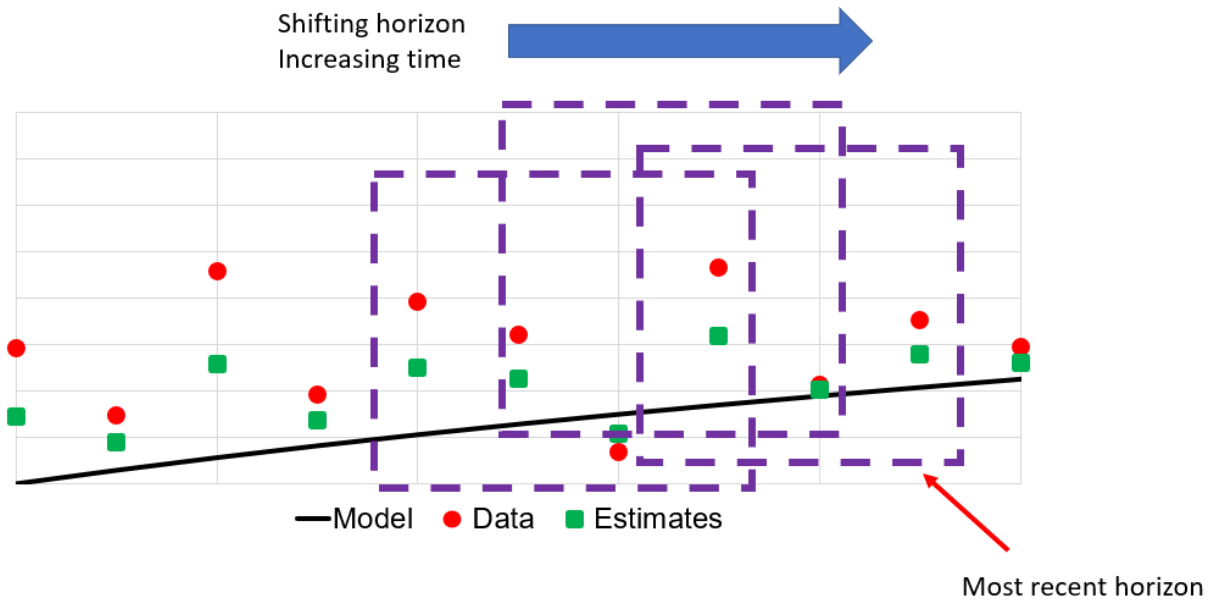

Figure 2.2 Shifting horizon window of MHE

The horizon window (dashed purple box) shown in Figure 2.2 uses a horizon length of 4 (i.e., N=4) and only utilizes the most recent measurements (red circles) available to the estimator. Using these specific data points and the process model, MHE derives a set of state estimates (green squares). Upon collection of new data, the horizon window advances forward by removing the oldest measurement from the window and replacing it with the new measurement.

Due to the flexibility of the MHE problem structure, a variety of alternative formulations can be found in the literature including parameter and state estimation forms (Kühl et al., 2011; Alexander et al., 2023). A particular challenge of using most MHE techniques is the significant increase in computational cost when compared to recursive techniques. This has motivated research into appropriate optimizer selection and horizon length selection (Rao & Rawlings, 2002; Thierry et al., 2018; Zavala & Biegler, 2009). Many optimizers can be used to solve this estimation problem including

sequential quadratic programming (SQP), modified SQP (*modSQP*) (He & Lima, 2020), IPOPT (Wächter & Biegler, 2006), meta-heuristic methods such as particle swarm optimization (PSO) (Zhou & Tan, 2009), or even combinations of different optimizers (e.g., PSO + SQP) (Kim & Lima, 2020).

The main limitation of using MHE over EKF is the associated increase in computational time due to repetitively solving online (often nonlinear) optimization problems instead of recursive calculations. For example, for a 2×2 nonlinear system MHE can have a 3-4 order of magnitude increase in computational time over traditional EKF estimation methods (Alexander et al., 2020). The specific increase in computational time depends on many factors including optimizer choice, problem complexity and size of the optimization problem, and horizon length used. Depending on the specific requirements of the process, this high computational time may not allow for MHE to be applied in an online manner as state estimates may not be readily available. In general, many of these estimation techniques have been developed for continuous processes operating at steady state, or transient processes shifting between steady states. Overall, there has been a lack of systematic algorithms developed or applied for semi-batch processes which do not have a steady state. As a result, this motivates Aim 1 of this dissertation, the application and development of specific state estimation algorithms for nonlinear semi-batch processes subject to plant-model mismatches.

2.2 Covariance Estimation

A limitation of both recursive and optimization-based approaches is having to supply accurate values of the model covariance matrix (Q) and measurement covariance

matrix (*R*) to obtain accurate approximations of the true process states. Since the development of the original KF algorithm, researchers have emphasized that poor approximations of these covariances can produce higher peak variances, slower convergence, instability, and overall suboptimal performance (Heffes, 1966; Nishimura 1967).

Estimation of the model and system noise covariances is a challenging endeavor and is most commonly conducted using ad-hoc techniques. Often control engineers will assume the *Q* and *R* matrices are diagonal in nature, which carries the assumption all sources of process and measurement noises are uncorrelated. A consequence of this assumption is a significant reduction in the number of variables that must be tuned by the control engineer. This more easily allows the process control engineer to simply modify the specific values of *Q* and *R* until acceptable estimation performance is achieved (Lima et al., 2013). Even with this simplification, these ad-hoc techniques cannot guarantee accuracy, can be time consuming to implement, and quickly become infeasible to manage for large nonlinear systems. These challenges have motivated the development of more systematic approaches for deriving covariance estimates.

Currently there is no de facto algorithm or systematic approach that has reached a consensus in the covariance estimation literature, as numerous methods have been proposed. These methods vary in terms of the required assumptions including linearity, number of system noises (e.g., full matrix estimation or diagonal elements only), and observability. Generally, these algorithms can be divided into 4 main categories: 1) correlation techniques, 2) maximum likelihood, 3) covariance matching, 4) Bayesian methods (Duník et al., 2017; Duník et al., 2018; Odelson et al., 2005).

Of these four methods, both Bayesian and maximum likelihood methods are commonly unpreferred due to their large computational time (Odelson et al., 2005). Bayesian methods employ a recursive joint estimation of the process states along with the covariance matrices using a nonlinear state estimator (Duník et al., 2018). Maximum likelihood methods are similar to Bayesian methods but conduct joint estimation of states and covariance matrices though maximization of a likelihood function (Duník et al., 2018; Kashyap, 1970). Covariance matching methods derive the covariance matrices by employing a statistical filter using the actual states and measurement estimation error statistics (Duník et al., 2018). Commonly cited algorithms using covariance matching include noise covariance matrices estimation with Gaussianity assessment (NEGA) (Duník et al., 2018; Duník et al. 2020*)* and adaptive limited memory filter (ALMF) (Myers & Tapley, 1976). Although NEGA has been applied to both linear time-invariant and linear time-varying systems, this class of covariance estimation technique has been shown to give biased estimates of the true covariances (Odelson et al., 2005). This leaves correlation techniques, which employ the innovation sequences of a linear estimation with a statistical assessment (Odelson et al., 2005). These methods were originally developed by Merha (1970) and Bélanger (1974) and were used as a starting point for the development of the ALS method.

ALS is a widely used covariance estimation technique and has gone through several revisions to build upon and resolve some concerns with Mehra's approach such as uniqueness and variance reduction. Originally, ALS was posed as a linear covariance estimation technique and generated more accurate estimates of the system and model covariances over Mehra's approach (Odelson et al., 2005). Overtime, ALS has been

further extended to nonlinear systems with EKF estimation (Rajamani et al., 2007) and nonlinear systems with MHE (Lima & Rawlings, 2011). From this point, ALS has been further modified and applied to weakly observable systems (Lima et al., 2013), batch processes (Rincón et al., 2014A), and semi-continuous (AKA, semi-batch) polymerization processes (Rincón et al., 2014B).

Despite ALS's proven track record and numerous successful applications, major algorithm development has stopped. The ALS package download available from Rawling's website from the University of California, Santa Barbara was last updated on December 5th, 2014, and requires using GNU Octave with Sundials toolbox (Rawlings, 2019). GNU Octave is a free high-level programming language mostly compatible with basic MATLAB functions (Eaton, n.d.; Eaton, 2012) but has not been widely disseminated in PSE applications. Figure 2.3 highlights the disparity between Google search results for GNU Octave and MATLAB.

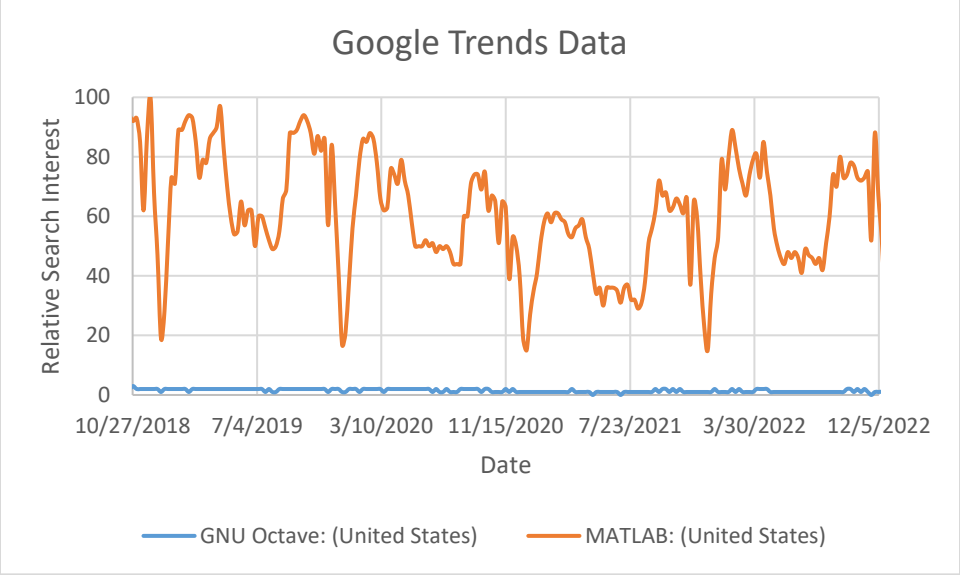

Figure 2.3 Google trends data for GNU Octave and MATLAB

The results shown in Figure 2.3 are generated from Google Trends (2023) and show the relative search interest for the two different coding languages. A value of 100 indicates the peak popularity of the search results, while a value of 50 indicates a search popularity of half the peak value. Overall, this search result demonstrates and supports the claim that GNU Octave popularity is not as high as MATLAB. As a result, the ALS toolbox currently available for download is rather out of date and unlikely to be used by the PSE community. Due to past successful applications of ALS, it has motivated the development of a MATLAB specific version and application to semi-batch processes considering plant-model mismatches in Aim 2 of this dissertation.

2.3 Dynamic Optimization Considering Process Economics

Optimization of chemical processes is a widespread field that can traditionally be divided into two main categories: steady-state optimization and dynamic optimization. Steady-state optimization is generally well understood by most process engineers and is commonly used to justify long term (i.e., days, weeks, years) changes to the plant through modification of the plant operating conditions or even capital projects by modifying topology. Costing and evaluating long-term capital projects for steady-state processes is a well-documented topic and can be solved using tools such as CAPCOST (Turton et al., 2018), process simulators (e.g., Aspen Capital Cost Estimator™ (AspenTech, n.d. A), Aspen Utilities™ (AspenTech, n.d. B), CHEMCAD's cost algorithms (Chemstations, n.d.)) or even internal company costing guidelines or vendor quotes. Changes to plant operating conditions generally fall under the purview of real-time optimization (RTO). RTO simply modifies setpoints throughout units or entire plants to

improve metrics such as process economics with constraints on product quality, production rate, safety operating regions, or other limits. RTO is a well-established technology and is commonly used in the petrochemical industry, paper industry, and is often a key part of many modern control systems (Câmara et al., 2016).

Dynamic optimization is a less defined area of research in the chemical process industry and can encompass different topics depending on the specific definition being used. For the purposes of this dissertation, dynamic optimization can be viewed as a strategy "that provides an optimal input trajectory using a dynamic model" (Krishnamoorthy et al., 2018) and includes algorithms such as DRTO and MPC (Krishnamoorthy et al., 2018; Kim & Lima, 2022). When considering MPC, most of the literature for DRTO is dedicated to improving the transient behavior of processes while transitioning between various steady states. As a result, DRTO and steady-state optimization approaches most commonly exclude processes that never reach steady-state and are always operating dynamically. A common example of such a process is a semi-batch reactor.

Semi-batch processes fall outside the scope of much of the existing literature and exhibit unique characteristics such as: inherent dynamic behavior, lack of steady states, mass accumulation/decumulation, and time dependent reactions. These characteristics make semi-batch reactor processes attractive for liquid phase reactions where selectivity is a key concern (Fogler, 2016). This dynamic behavior, controlled inputs, and lack of steady states produce a challenging economic optimization problem that has not been thoroughly addressed in the literature. Recent publications such as Kummer (2020) and Sass (2022) have addressed some control aspects of semi-batch processes using

various techniques, without focusing on maximizing the profitability. This lack of emphasis on economic optimization of semi-batch processes through optimal control/DRTO is the motivation behind Aim 3 of this dissertation.

Chapter 3 State Estimation Applied to Nonlinear Semi-Batch Processes

In this chapter, both recursive and optimization-based nonlinear state estimation techniques are applied to a semi-batch GOS process where not all process states can be measured using online techniques. As a result, this requires the use of estimation algorithms to conduct process monitoring or process optimization. Noise is introduced into the system by varying the kinetic parameter values and applying white noise to synthetically generate measurements. The work presented in this chapter was published as part of Alexander et al. (2023).

This chapter begins by providing a brief overview of the GOS market, the process model selected, and the procedure used to generate simulated data for the estimation studies. Then, various recursive estimation techniques are tested and evaluated in terms of accuracy and robustness. As part of this chapter, a preliminary covariance estimation technique is also discussed for this class of estimation algorithms which is further elaborated in Chapter 4. Finally, optimization-based state estimation techniques are also tested and evaluated to determine their effectiveness on semi-batch processes.

3.1 Galacto-oligosaccharide Overview

Although a variety of chemical or biochemical processes could have been selected as the case study for this work, the synthesis of GOS via the transgalactosylation of lactose using β-galactosidase was chosen due to several main reasons: 1) health benefits, 2) application of green chemistry principles, 3) increasing

market demand, 4) unmeasurable states requiring state estimation, 5) potential improvement from operation in a semi-batch form.

GOS refers to a class of value-added food products that are traditionally derived from lactose to form a series of galactose and glucose-based products (often galactobiose, a mixture of tri- and tetrasaccharides) (Yin et al., 2017; Frenzel et al., 2015). These compounds have been associated as having health benefits for intestinal microbiota and barrier functions, which have shown positive effects for mineral absorption, weight management, and decreased carcinogenesis on intestinal microbiota (Lamsal, 2012). Beyond these health benefits, GOS has demonstrated stability in adverse temperature and acidic conditions, thus making it an excellent food additive in dairy products (Catenza & Donkor, 2021).

Lactose-derived GOS products leverage green chemistry principles by remediating the waste disposal problem of dairy farms. During the production of cheese, dairy farms often produce lactose rich waste (i.e., whey permeate) which is problematic to dispose of and has few avenues for conversion to a value-added product (Albayrak & Yang, 2001). This whey byproduct has a high chemical oxygen demand (COD) and biochemical oxygen demand (BOD) and cannot be disposed of in municipal sewers, dumped onto land due to soil degradation, or discharged into water without risking damage to aquatic life (Yadav, 2015). Although unwanted whey production is not new to the dairy industry, currently an estimated 180 to 190 million tons/year are produced and this is likely to increase as whey generation increases at approximately 2% per year (Yadav, 2015). Currently, water from dairy processing is one of the largest sources of industrial food wastewater (Slavov, 2017). The production of GOS from lactose is thus

an excellent example of a potential application of green chemistry principles to bioprocess manufacturing.

Outside of the green chemistry motivations for this process, the market demand for GOS is strong with a proven worldwide consumer base. GOS containing products have been sold commercially in Japan and Europe since the 20th century as a prebiotic food additive (Wang et al., 2021). These products have helped GOS to reach an estimated market value of 1.01 billion USD with a product demand of 175.7 kilo tons in 2020 (Wang et al., 2021). Looking forward, the GOS market is expected to experience a compound annual growth rate (CAGR) of 7.5% through 2032 (Future Market Insights, 2020). As a result, there is an excellent opportunity to promote and continue the widespread adoption of this process.

Finally, this process poses significant challenges in terms of process monitoring as not all process states are able to be directly measured. The monomer building blocks that comprise these GOS products are galactose and glucose which are stereoisomers that differ exclusively in the positioning of the hydroxyl group on the 4th carbon. In turn, this similarity creates some difficulty when trying to isolate and differentiate these monomers (Sturgeon, 1990). As these compounds react to form higher-order saccharides, it becomes very difficult to distinguish the specific product distributions using offline or online analytical techniques.

In terms of offline analytical methods, High-performance Anion-Exchange Chromatography with Pulsed Amperometric Detection (HPAEC-PAD) with gradient elution is an effective chromatographic method that has high resolution for different

carbohydrates with the main drawback being an analysis time of 75 minutes or longer (Rodriguez-Colinas et al., 2011). High Performance Liquid Chromatography (HPLC) using Refractive Index detector (RI) and isocratic elution, is another more widespread offline technique due to its simplicity, lower analysis time of around 20 min, and ability to uniquely distinguish the concentration of galactose, glucose, all combined disaccharide isomers, trisaccharide isomers, and all combined tetrasaccharide isomers (Schultz et al., 2021).

For online process monitoring, there have been attempts to apply Fourier transformed mid-infrared (FTIR) (Rico-Rodriguez et al., 2021) and UV spectrophotometry (Dias et al., 2009) techniques combined with chemometrics to measure the concentration of the GOS products, galactose and glucose. These techniques are subject to a high amount of noise, poor calibration, and other factors that would reduce the accuracy of their associated measurements. Furthermore, these techniques are currently unable to measure the specific disaccharide, trisaccharide and tetrasaccharide isomer concentrations and instead are only able to measure the total concentration of the disaccharide, trisaccharide and tetrasaccharide isomers. Although spectrometric analysis may be able to determine these concentrations, the available HPLC reference data is unable to determine these specific concentrations. In essence, it is impossible to measure each component concentration using offline or online analytical techniques given the current technology available. As a result, the most convenient method for deriving the true process states is to apply a state estimator.

3.1.1 GOS Model Used

The GOS model used in this work is derived from Schultz et al. (2021) as it is one of the most recent kinetic models published in the literature. This model is unique as it incorporates enzyme deactivation and the formation of galactobiose, tri-, and tetrasaccharides into one cohesive model. Furthermore, unlike many of the existing models in the literature, the model differentiates between purely galactose rich tri- and tetrasaccharides compounds while striking a careful balance between complexity, accuracy, and parameter correlation to prevent model overfitting (Schultz et al., 2021). In literature, this model was published in a batch form but was converted to a dynamic continuous stirred-tank reactor (CSTR) form as shown below as Equations 3.1-3.15. Additionally, a pictorial representation of the dynamic process is shown in Figure 3.1.

$$\frac{dlac}{dt} = e\left(\frac{k_H}{K_{MH}} tri - \frac{k_{cat}}{KM} lac - \frac{k_T}{K_{MT}} \gamma\, lac\right) + \frac{u_{lac} C_{lac}}{V} - \frac{u_{in} lac}{V} \tag{3.1}$$

$$\frac{dglu}{dt} = e\frac{k_{cat}}{K_M} lac - \frac{u_{in} glu}{V} \tag{3.2}$$

$$\frac{dgal}{dt} = e\left(\gamma\left(k_{cat'} - \frac{k_T}{K_{MGal}} gal\right) + \frac{k_H}{K_{MH}} glb\right) - \frac{u_{in} gal}{V} \tag{3.3}$$

$$\frac{dtri}{dt} = e\left(\gamma \frac{k_T}{K_{MT}} (lac - tri) + \frac{k_H}{K_{MH}} (tet - tri)\right) - \frac{u_{in} tri}{V} \tag{3.4}$$

$$\frac{dtet}{dt} = e\left(\gamma \frac{k_T}{K_{MT}} tri - \frac{k_H}{K_{MH}} tet\right) - \frac{u_{in} tet}{V} \tag{3.5}$$

$$\frac{dglb}{dt} = e\left(\gamma k_T \left(\frac{gal}{K_{MGal}} - \frac{glb}{K_{MT}}\right) - \frac{k_H}{K_{MH}} (glb - trig)\right) - \frac{u_{in} glb}{V} \tag{3.6}$$

$$\frac{dtrig}{dt} = e(\gamma \frac{k_T}{K_{MT}} (glb - trig) + \frac{k_H}{K_{MH}} (tetg - trig) - \frac{u_{in} trig}{V} \tag{3.7}$$

$$\frac{dtetg}{dt} = e\left(\gamma \frac{k_T}{K_{MT}} trig - \frac{k_H}{K_{MH}} tetg\right) - \frac{u_{in} tetg}{V} \tag{3.8}$$

$$\frac{dE}{dt} = -k_e E + \frac{u_{enz} C_{enz}}{V} - \frac{u_{in} E}{V} \tag{3.9}$$

$$\frac{dv}{dt} = u_{in} \tag{3.10}$$

$$u_{in} = u_{enz} + u_{lac} \tag{3.11}$$

$$\gamma = \frac{\frac{k_{cat}}{K_M} lac + \frac{k_H}{K_{MH}} a}{k'_{cat} + \frac{k_T}{K_{MGal}} gal + \frac{k_T}{K_{MT}} b} \tag{3.12}$$

$$a = glb + tri + trig + tet + tetg \tag{3.13}$$

$$b = lac + glb + tri + trig \tag{3.14}$$

$$e = \frac{E}{1 + \frac{Gal}{K_I} + \frac{Lac}{K_M} + \frac{a}{K_{MH}} + \gamma\left(1 + \frac{Gal}{K_{MGal}} + \frac{b}{K_{MT}}\right)} \tag{3.15}$$

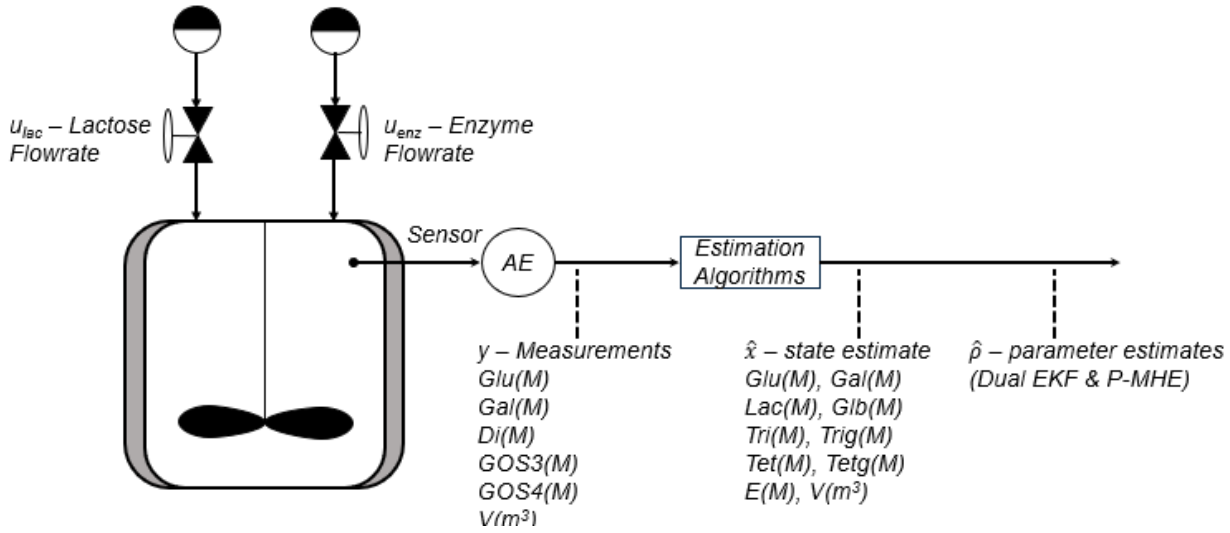

Figure 3.1 Schematic of dynamic GOS process

Equations 3.1-3.10 characterize the rate of change of the 10 state variables and are influenced by independent feed streams into the reactor as shown in Figure 3.1.

Lactose (u_{lac}) and enzyme flowrates (u_{enz}) of constant lactose (C_{lac}) and enzyme feed concentrations (C_{enz}) are independently regulated to control lactose concentration (*Lac*) and active enzyme concentration (*E*) throughout operation of the semi-batch process. Throughout the batch, lactose is broken down to increase the glucose (*Glu*) and galactose (*Gal*) concentrations. These monomer building blocks subsequently influence the formation of and concentration of galactobiose (*Glb*), glucose rich trisaccharides (*Tri*), glucose rich tetrasaccharides (*Tet*), purely galactose rich trisaccharides (*Trig*), and purely galactose rich tetrasaccharides (*Tetg*). Over time, *E* decreases and is maintained by feeding fresh active enzyme in the reactor. Additionally, if a semi-batch operation is assumed than reactor volume (*V*) changes dynamically subject to the system inputs (u_{lac} and u_{enz}).

Nonlinearities are introduced into the process model described by the differential equations through the presence of the parameters *γ*, *a*, and *b*, which involve most process states (excluding *Glu*, *Gal*, *E* and *V*). Variable *e* in the model is the free enzyme concentration (not linked or complexed with substrates or products) which is evaluated from the mass balance on the total enzyme load. The 10 kinetic parameters associated with using *Kluyveromyces lactis* (*K. lactis)* β-galactosidase and their nominal values (regressed at 40 °C and pH 7) can be found in Table 3.1 (Schultz et al., 2021).

Table 3.1 Values of kinetic parameters for *K. lactis* β-galactosidase (reproduced from Schultz et al., 2021)

Kinetic Parameter (unit)	Value
k_{cat} (min^{-1})	$3.526 \cdot 10^{7}$
$k_{cat'}$ (min^{-1})	$9.943 \cdot 10^{9}$
K_{MH} (mol/L)	$3.345 \cdot 10^{-6}$
K_{MT} (mol/L)	$1.758 \cdot 10^{-6}$
k_H (min^{-1})	$2.876 \cdot 10^{3}$
k_T (min^{-1})	$1.439 \cdot 10^{4}$
K_I (mol/L)	$9.405 \cdot 10^{-6}$
K_M (mol/L)	$2.586 \cdot 10^{-2}$
K_{MGal} (mol/L)	$2.433 \cdot 10^{-6}$
k_E (min^{-1})	$1.651 \cdot 10^{-4}$

One important caveat to this GOS model not encountered in this specific implementation is the lactose solubility limit and its temperature dependency. Like most bioproducts, the solubility of lactose increases with temperature and can vary from 10 wt% to around 65 wt% across a temperature range of 0 °C to 100 °C. In this work, a reaction temperature of 40°C was assumed (consistent with the kinetic parameter regression temperature) which correlates to a lactose solubility limit of approximately 25 wt% (0.88 M) (DFE Pharma, 2006). This constraint was avoided by feeding fresh lactose at a concentration less than the solubility limit. In real GOS processes this solubility limit may not be a concern as lactose fed into the reactor as a solid could continuously enter the liquid phase to maintain operation at the lactose solubility limit. However, the current GOS model has not been designed for solid-liquid phase equilibrium but could be further developed to address this specific form of operation.

As stated in the previous section, not all process states are measurable for this process. Figure 3.1 depicts the 6 measurements that are available for this process: total disaccharide concentration (*Di*), *Glu, Gal,* total trisaccharide concentration (*GOS3*), and total tetrasaccharide concentration (*GOS4*), and *V.* Although *Di*, *GOS3*, and *GOS4* do not directly measure state variables, they are result of the joint concentration of several process states defined in Equations 3.16-3.18 and can be used in conjunction with the process model to infer the individual state concentrations (*Glb*, *Lac*, *Tri*, *Trig*, *Tet*, *Tetg*). Only state *E* is unable to be measured directly since it is unclear if it is possible to have a good estimate of the enzyme load in this system using UV or NIR spectra without extensive laboratory experiments using various enzyme concentrations and careful calibration for this variable.

$$Di = glb + lac \tag{3.16}$$

$$GOS3 = tri + trig \tag{3.17}$$

$$GOS4 = tet + tetg \tag{3.18}$$

3.1.2 Simulated Data Generation

Due to cost limitations and limited amounts of data, it is often challenging to conduct robust state estimation studies using real bioprocess data. As a result, the procedure shown below in Figure 3.2 is used to generate synthetic data for mimicking the real process data.

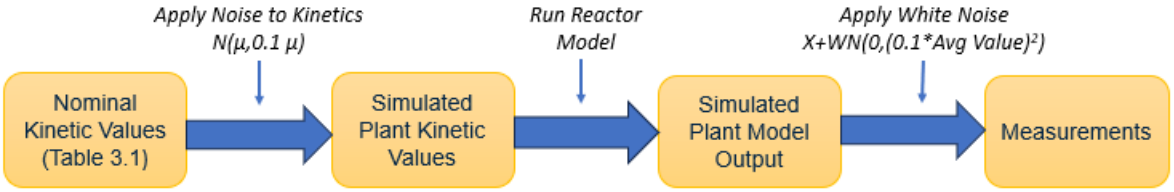

Figure 3.2 Procedure for generating synthetic data for bioprocess application

Figure 3.2 begins with varying all kinetic parameters independently by applying a normal distribution to the nominal (true) kinetic values outlined in Table 3.1. These randomized kinetic parameters become the simulated plant kinetic values and reflect disturbances in pH, temperature, and other process conditions that modify the underlying enzymatic reactions. This set of kinetic parameter values is used in conjunction with the reactor model described in the previous section to produce the simulated plant model output. Upon derivation of this data set, Equations 3.16-3.18 and a Gaussian white noise distribution (*WN*) are applied to generate the simulated data. Table 3.2 supplies the average state values used for the Gaussian white noise distributions, which are derived offline by running extensive simulations employing the reactor model with random control inputs thousands of times and finding the average value of each measurement across all sampled times and iterations. Overall, this framework creates a plant-model mismatch and applies a Gaussian white noise to the plant model to simulate real process measurements.

Table 3.2 Average state values considered

Measurement	Average State Value
Disaccharide Concentration (M)	0.0197
Glucose Concentration (M)	0.00973
Galactose Concentration (M)	0.00714
GOS 3 Concentration (M)	0.00109
GOS 4 Concentration (M)	$5.53 \cdot 10^{-5}$
Volume (m^3)	$2.90 \cdot 10^{-3}$

State estimation is essential for this process as any disturbance that affects the enzymatic model will eventually propagate throughout the operation of the fed-batch reactor and influence the amount of GOS produced, component concentrations, and the active enzyme concentration. To visualize and show this propagation occurring, a Monte Carlo analysis is carried out by running the procedure outlined in Figure 3.2. Figure 3.3 shows the distribution of all kinetic parameters used in the Monte Carlo analysis, and Figure 3.4 depicts the distribution of selected component concentrations and total GOS moles synthesized from running the simulated plant model.

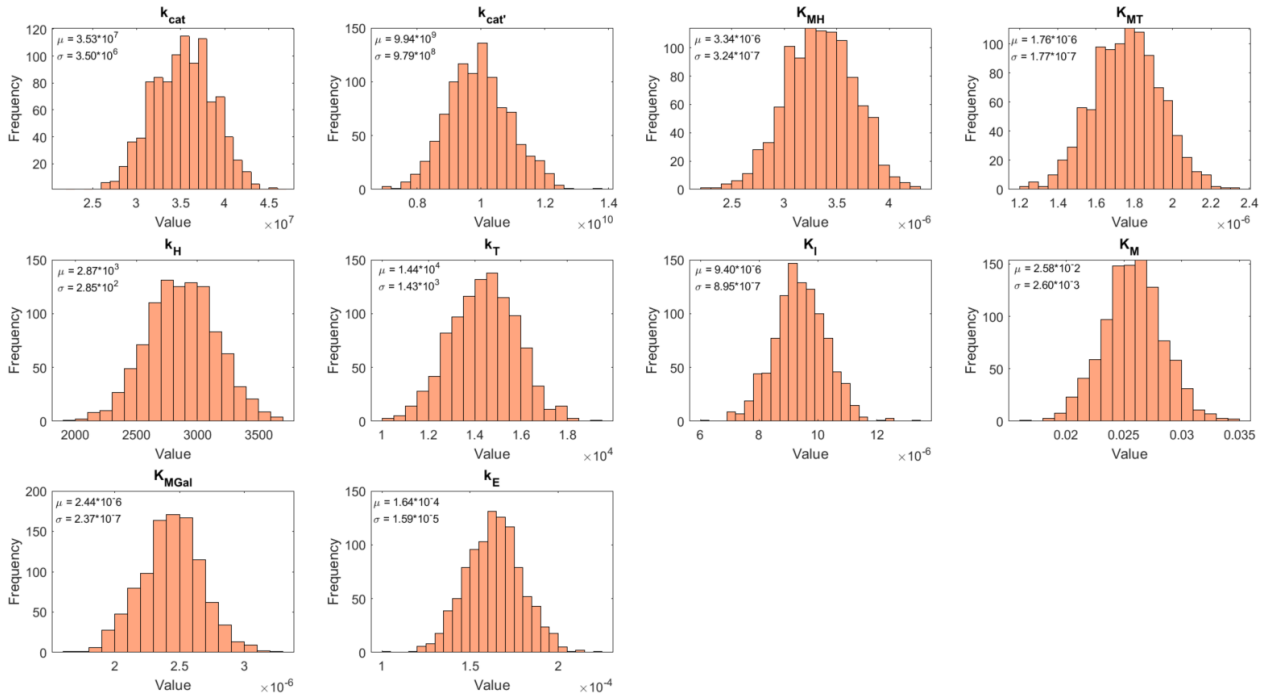

Figure 3.3 Distribution of kinetic parameters considered

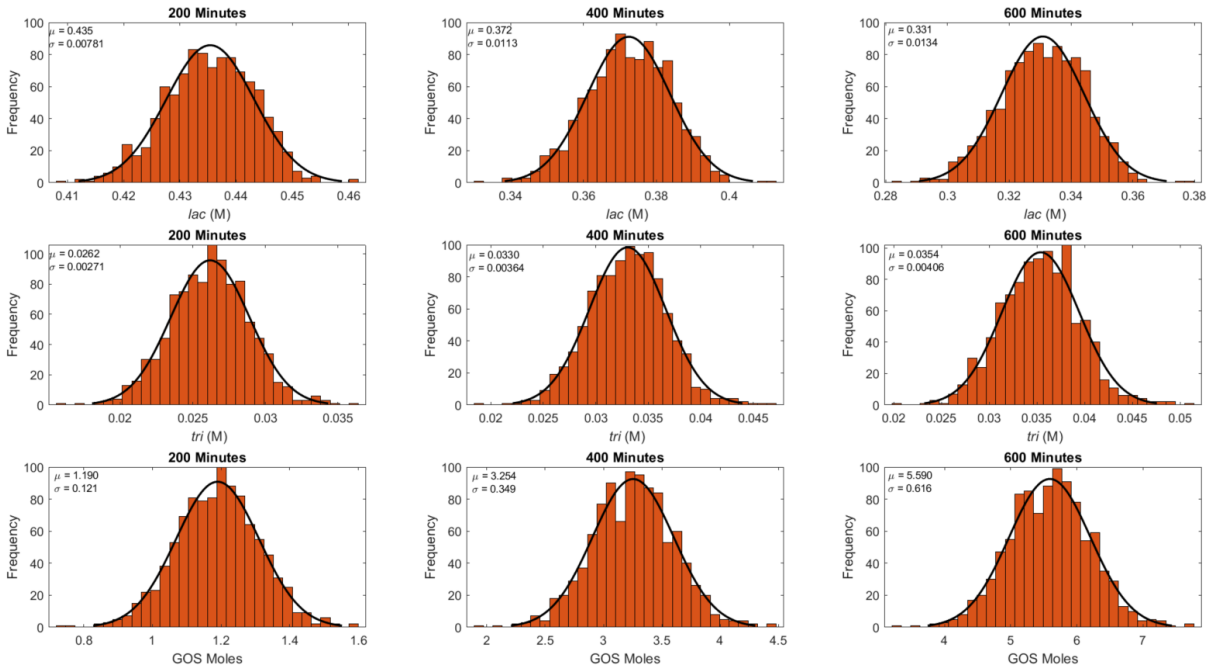

Figure 3.4 Distribution of selected concentrations and GOS moles produced

The distributions shown in Figures 3.3 and 3.4 are generated using 1000 simulations, for a constant lactose to enzyme ratio of 7, a constant feeding rate of $2 \cdot 10^{-4}$ m^3/min, and the parameters outlined in Table 3.3. The kinetic parameter distributions shown in Figure 3.3 follow the expected Gaussian behavior and have a mean value identical to the nominal kinetic values. Due to the dynamic nature of this process, Figure 3.4 tracks *lac* (top row), *tri* (middle row), and the number of GOS moles (bottom row) distributions at three different times: 200 minutes (left column), 400 minutes (middle column), and 600 minutes (right column). The number of GOS moles is defined below in Equation 3.19 and is a useful metric for characterizing the performance of this GOS process. Across all the plots shown in Figure 3.4, the distribution of kinetic parameters ultimately results in a Gaussian process for the examined outputs with time varying means and variances. Ultimately this demonstrates that any uncertainty or disturbance in the kinetic parameters, propagates through the reaction system model and leads to uncertainty in the process states which can be mitigated by the application of a state estimator.

$$GOS\ moles = (glb + tri + trig + tet + tetg) \cdot V \tag{3.19}$$

Table 3.3 Reactor concentration parameters employed

Constant	**Value**
$C_{lac\ init}$ – Initial Lactose Concentration	0.652 (M)
$C_{enz\ init}$ -Initial Enzyme Concentration	0 (M)
C_{lac} – Feeding Lactose Concentration	0.652 (M)
C_{enz} – Feeding Enzyme Concentration	$3.63 \cdot 10^{-7}$ (M)

3.2 EKF-Based Estimation Approach

Due to the nonlinear nature of the GOS model, linear estimation filters were not considered for implementation. Instead, the first estimation algorithm implemented into the GOS case study was the classical EKF formulation as shown below in Equations 3.20-3.28, which is adapted from Rawlings et al. (2022).

$$\hat{x}_0^- = \bar{x}_0 \tag{3.20}$$

$$P_0^- = Q \tag{3.21}$$

$$\hat{x}_{k+1}^- = f(\hat{x}_k, u_k) \tag{3.22}$$

$$P_{k+1}^- = \bar{A}_k P_k \bar{A}_k' + Q \tag{3.23}$$

$$\hat{x}_k = \hat{x}_k^- + L_k(y_k - h(\hat{x}_k^-)) \tag{3.24}$$

$$L_k = P_k^- \bar{C}_k'(R + \bar{C}_k P_k^- \bar{C}_k')^{-1} \tag{3.25}$$

$$P_k = P_k^- - L_k \bar{C}_k P_k^- \tag{3.26}$$

$$\bar{A}_k = \frac{\delta f(x,u)}{\delta x}\Big|_{(\hat{x}_k, u_k)} \tag{3.27}$$

$$\bar{C}_k = \frac{\partial h(x)}{\partial x}\Big|_{\hat{x}_k^-} \tag{3.28}$$

The EKF algorithm follows the basic procedure outlined in Figure 2.1 and is initialized using Equations 3.20 and 3.21. The initial guesses are used in the correction step of the algorithm as shown in Equations 3.24-3.26. Within this correction step, Equation 3.25 is used to calculate the Kalman filter gain (L_k) (AKA optimal estimator gain). An important distinction with this filter formulation is the ability to use a nonlinear

equation to relate the state estimate forecast to the measurements via a nonlinear system function ($h(\hat{x}_k^-)$) as shown in Equation 3.24. This nonlinear function is only applied in Equation 3.24 and must be linearized using Equation 3.28 to derive the linearized controllability matrix ($\bar{C}_k$) required in Equations 3.25 and 3.26. Upon completion of this correction step, the forecast step is conducted using Equations 3.22 and 3.23

Due to the flexibility of EKF, the forecast of the state estimate is conducted using the nonlinear reactor model ($f(\hat{x}_k, u_k)$) directly with the state estimates and the control inputs as shown in Equation 3.22. The forecast of the covariance matrix is conducted in Equation 3.23 and done using the linearized system matrix ($\bar{A}_k$) which is derived using Equation 3.27.

The various states and measurements of the GOS model vary in order of magnitude from 10^{-8} to 10^{-1} and pose significant challenges for the EKF. In general, these large differences tend to cause EKF instability and more difficulty when tuning the filter. As a result, in this work state values and measurements are normalized for use in the algorithm. Normalization is conducted by following the procedure outlined in Figure 3.2 and calculating the mean value of the state variables and measurements. These mean values are used to form the state estimate normalization vector and the measurement normalization vector, which are subsequently divided and multiplied where needed to convert the model forecast and raw measurements to the normalized values required in the EKF.

3.2.1 Application of Direct Optimization for EKF

As noted in Section 2.2, poor approximation of the model and measurement covariances will produce suboptimal estimator performance. To remediate this challenge, a basic optimization-based covariance estimation method known as DO is used in this section of the dissertation. DO was chosen for several reasons including its level of simplicity for the user, the ability to be conducted offline, incorporation of real process data, and minimization of estimator error (Rincón et al., 2013; Rincón et al., 2014A). The standard DO formulation is shown below as Equations 3.29-3.32 and is adapted from Rincón et al. (2014A). A schematic of the standard DO formulation is shown in Figure 3.5.

$$(Q,R) = argmin_{(Q_{DO},R_{DO})} \sum_{j=1}^{J} \sum_{k=1}^{K} \left(y_k(j) - \hat{x}_k(j)\right)^T \left(y_k(j) - \hat{x}_k(j)\right) \tag{3.29}$$

$$Q_{DO} = diag[w_1, w_2, \dots, w_q] \tag{3.30}$$

$$R_{DO} = diag[v_1, v_2, \dots, v_r] \tag{3.31}$$

s.t. EKF and process model

$$(Q,R) = argmin_{(Q_{DO},R_{DO})} \sum_{j=1}^{J} \sum_{k=1}^{K} \left(y_k(j) - \bar{C}_k\hat{x}_k(j)\right)^T \left(y_k(j) - \bar{C}_k\hat{x}_k(j)\right) \tag{3.32}$$

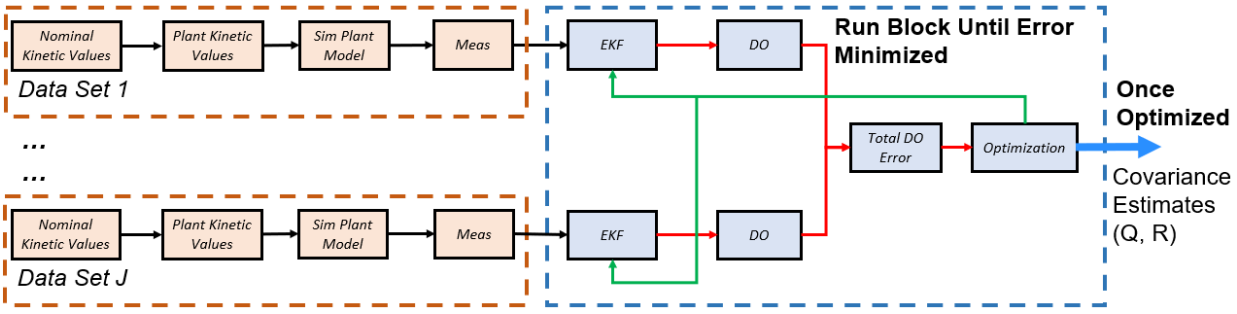

Figure 3.5 Standard DO framework using simulated data from several simulations

Equation 3.29 is the standard DO objective function and defines the estimation error as the difference between measurements and the state estimates (generated from the EKF) across *K* time steps and *J* number of fed-batches with the assumption that all states are directly measurable. For clarity, in the term $y_k(j)$, subscript *k* corresponds to the specific discrete time and parenthesis *j* corresponds to the specific fed batch data set. The estimation error is minimized by varying the process model and measurement covariances defined in Equations 3.30 and 3.31, respectively. Figure 3.5 shows that each data set is associated with a specific EKF implementation that receives updated covariances from the optimization algorithm. Upon rerunning all EKFs with these updated covariances, the resulting state estimates are used to calculate the objective function. This closes the optimization loop and is repeated until the optimizer converges or reaches some predefined stopping criteria.

In this DO implementation, it is assumed that all noises in the system are uncorrelated, which limits the optimization to only the diagonal components of the process and measurement noise covariance matrices. This drastically reduces the size of the optimization problem from $\sum_{i=1}^{q} i + \sum_{i=1}^{r} i$ variables to $q + r$ variables. Not every element in matrix Q and R can/should be directly optimized as the covariance matrices are symmetric (e.g., (1,2) = (2,1)). In this system, this produces a 16 variable optimization problem as matrix *Q* is 10x10 (*q*=10) and matrix *R* (*r*=6) is 6x6. Although this simplification eliminates the covariance relationships between states and measurements (i.e., off diagonal cells in matrices *Q* and *R*), this is a commonly accepted practice in estimation applications (Bolognani et al., 2003). Feasibility is guaranteed by constraining the model noise (*w*) and measurement noise (*v*) to have values greater than 0. An upper

value constraint of 10 is applied here to all *w* and *v* terms to facilitate convergence of the DO algorithm but this limit can be specified as any positive value. Due to the lack of independent measurements for all process states, Equation 3.32 is modified here with the measurement matrix $\bar{C}_k$ to allow for the di-, tri-, and tetrasaccharide measurements to be used, producing an alternative form of the standard DO algorithm.

The standard DO algorithm in literature employs a measurement bias (i.e., only incorporates data from measurements) and is modified in this work. This modified form employs the measurements, model outputs, and state estimates concurrently to reduce overfitting of the measurements. This modified DO algorithm is shown in Equation 3.33 and is combined with the EKF using the framework shown in Figure 3.6. In Equation 3.33, x_{model} refers to the model output block (*Model Output*) of Figure 3.6. Visually, Figure 3.6 depicts how the standard DO algorithm is modified by generating simulated data in parallel to the measurement data. This model data is directly fed into the DO block of the algorithm and is used to reduce the bias from solely using the measurements.

$$(Q,R) = argmin_{(Q_{DO},R_{DO})} \sum_{j=1}^{J} \sum_{k=1}^{K} \left(y_k(j) - \bar{C}_k \hat{x}_k(j)\right)^T (y_k(j) - \bar{C}_k \hat{x}_k(j)) + \sum_{j=i}^{J} \sum_{k=1}^{K} (x_{model\,k}(j) - \hat{x}_k(j))^T (x_{model\,k}(j) - \hat{x}_k(j)) \quad (3.33)$$

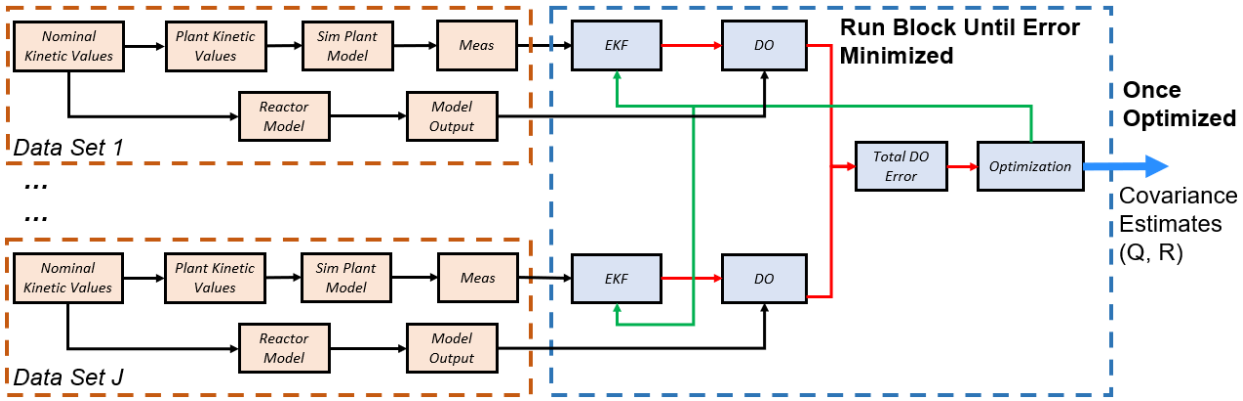

Figure 3.6 Modified DO framework using simulated data from several simulations

Figure 3.6 can be broken down into two main segments: generation of the simulated data sets using a Monte Carlo approach (dashed brown boxes) and running the DO/EKF algorithm (blue dashed boxes). The generation of simulated data sets follows the procedure outlined in Figure 3.2 and applies a uniform distribution to vary the reactor manipulated variables to expand the range of possible operating conditions. Any number of data sets can be used in this framework, but all results presented in this chapter of the dissertation use 50 random data sets (J=50). Upon generation of these data sets, the DO algorithm is initialized by running the EKF with initial feasible approximations of the measurement and process model covariances to produce the state estimates. This is proceeded by calculating the DO error for each simulation which are subsequently summed to calculate the total DO error. This total DO error is sent to the optimization algorithm (*fmincon* in MATLAB) to update the process and measurement noise covariances, subject to the feasibility constraints. These updated covariances are fed back into the estimation algorithm, closing the DO algorithm.

This estimation, DO error calculation, and optimization cycle is iterated until the optimization algorithm reaches convergence or some other predefined stopping criteria. In this work, the default *fmincon* settings except for 5000 max function evaluations and 5000 max iterations (both larger than the default settings) are used. It is important to note that across all iterations of the DO algorithm, the data sets remain constant (to represent available historic process data).

3.2.2 EKF and Modified DO Case Study

To validate and test this framework, a series of randomized case studies are carried out using separate synthetic measurements, the derived covariances, and an EKF. Figure 3.7 shows the EKF estimation performance in terms of the state variables using a selected random data set with its corresponding set of manipulated variable values, and the covariance values derived employing the process outlined in the previous section.

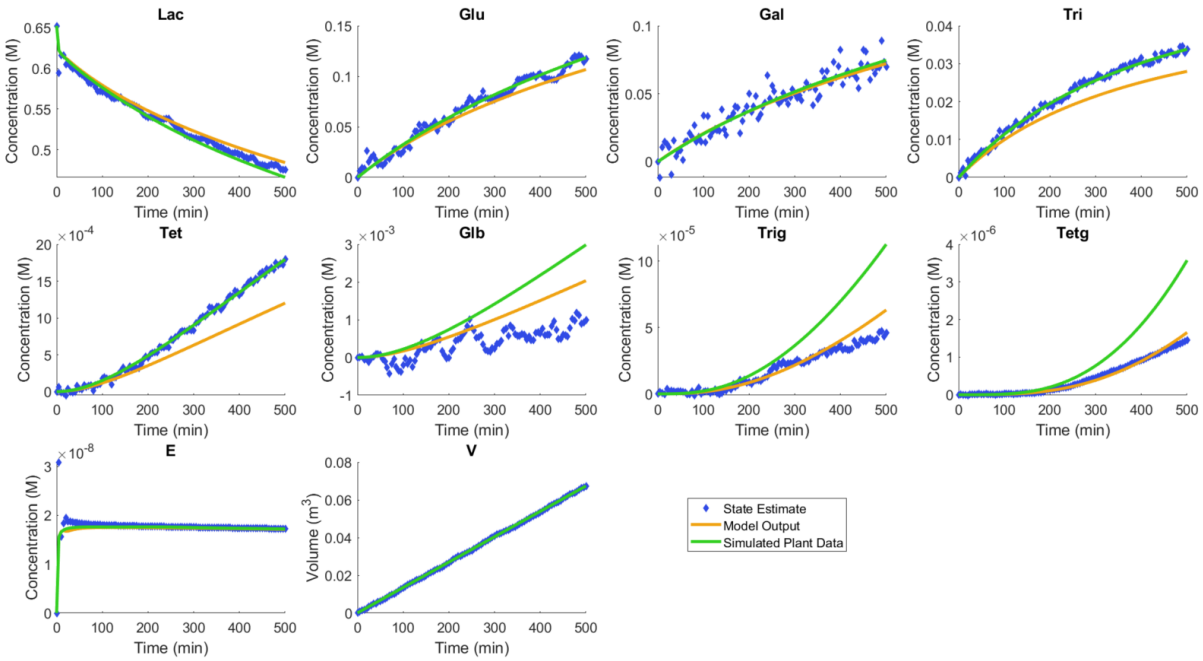

Figure 3.7 EKF estimation performance for state variables using Modified DO

To assist in visualizing the estimator performance for the state variables, Figure 3.7 shows the model output (generated from the nominal kinetic parameters), the simulated plant data (generated from the "true" but unknown kinetic parameters), and the obtained state estimates. Typically, for state estimation applications, the state estimates should lie between the model output and measurements (simulated plant data + noise) as the true process states are likely to lie within this region. When using the modified DO algorithm, many of the state variables lie within the model output and simulated plant data bands. States *Lac*, *Glu*, *Tri*, and *Tet* converge to the simulated plant data while state *Tetg* converges to the model output. States *Glb* and *Trig* fall outside of these bands, but generally follow a similar trajectory with converged state estimate values. Due to the lack of unique/direct measurements of states such as *Glb* and *Trig*, it is challenging to derive accurate estimates of these specific states. These states are

measured concurrently with their reciprocal isomers (*Lac* and *Tri*) which may be of much larger concentrations to form the Disaccharide and GOS3 measurements. As a result, there is a bias towards their reciprocal isomers, resulting in worse estimation results for *Glb* and *Trig*. State *Gal* oscillates around the model output and simulated plant data bands; thus, not producing a series of smooth states estimates. Figure 3.8 shows the estimation performance in terms of the outputs and measurements for this case study.

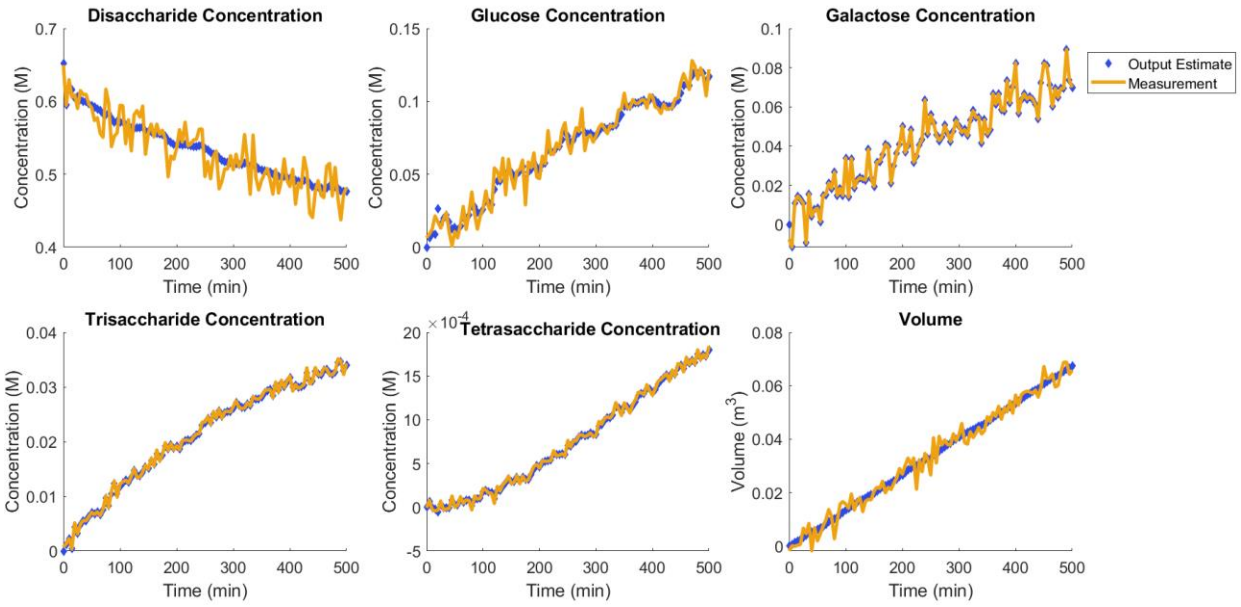

Figure 3.8 EKF performance results for outputs using Modified DO

The results shown in Figure 3.8 are generated using the same data and estimation results from Figure 3.7. For the di, tri, and tetrasaccharide concentration measurements, Equations 3.16-3.18 are used to generate the output estimate for these measurements (i.e., reconstruct measurements from constituent isomer components). When applying the modified DO algorithm, the output estimates do not strictly follow the measurements. Instead, most output estimates (*Di-*, *Tri-*, *Tetrasaccharide*, *Glucose*, and *Volume*) lie

within the peaks and valleys of the measurement data and maintain a relatively smooth profile. Only the *Galactose* output estimate oscillates synchronously with the measurement. These estimation results can be explained by consulting the covariance values obtained using this framework. The average measurement covariance value is approximately 4.8 while the average model covariance value is 2.9. These larger measurement covariance values result in less bias towards the measurements and result in a better balance between the model forecast and correction stages of the EKF algorithm. Overall, by applying this modified DO framework it is possible to filter out a large portion of the measurement noise while obtaining accurate state estimates for many of the state variables.

3.3 Dual EKF-Based Estimation Approach

The standard EKF implementation does not consider parameter uncertainty in its formulation and instead solely relies upon using the nominal kinetic parameters when conducting the state estimate forecast. As a result, the model forecast step of the EKF algorithm will always introduce estimation error as there is a chronic mismatch between the nominal kinetic parameters and the true kinetic values from the plant. This mismatch is not resolved in the traditional EKF algorithm implemented and produces the state estimation results shown above where it is challenging to obtain accurate state estimates for all process states. To improve the robustness of the estimation algorithm, a modified form of the EKF known as the Dual EKF is applied to simultaneously solve the kinetic parameter and the state estimation problems (Sun et al., 2008).

In the estimation literature, there are many publications involving joint parameter and state estimation using modified formulations of the EKF such the extended Kitanidis Kalman Filter (Varshney et al., 2019) and the Dual EKF (Sun et al., 2008). The specific Dual EKF used in this work has already been applied to nonlinear models of biochemical networks and can be readily implemented by making a few changes to the existing EKF implementation (Sun et al., 2008). The specific Dual EKF formulation used is shown below in Equations 3.34–3.42.

$$\hat{z}^{-}(0) = \bar{z}_0 \tag{3.34}$$

$$P^{-}(0) = \psi_0 \tag{3.35}$$

$$\hat{z}_{k+1}^{-} = f(\hat{z}_k, u_k) \tag{3.36}$$

$$P_{k+1}^{-} = \bar{A}_k P_k \bar{A}_k' + \psi_k \tag{3.37}$$

$$\hat{z}_k = \hat{z}_k^{-} + L_k[y_k - h(\hat{z}_k^{-}, u_k)] \tag{3.38}$$

$$L_k = P_k^{-} \bar{C}_k'[\bar{C}_k P_k^{-} \bar{C}_k' + R_k]^{-1} \tag{3.39}$$

$$P_k = P_k^{-} - K_k \bar{C}_k P_k^{-} \tag{3.40}$$

$$z = \begin{bmatrix} x \\ \rho_{kin} \end{bmatrix} \tag{3.41}$$

$$\psi = \begin{bmatrix} Q & 0 \\ 0 & \theta \end{bmatrix} \tag{3.42}$$

Once again, this recursive estimation technique follows the procedure outlined in Figure 2.1 and modifies the standard EKF by augmenting the state vector and the model covariance with parametric information. The initialization of the Dual EKF is conducted

in Equations 3.34 and 3.35 and remains largely unchanged from the standard EKF as the state vector and model covariance matrix are simply augmented with the kinetic parameters (ρ_{kin}) and their covariance matrix (θ) to form the augmented state vector (z) and augmented model covariance matrix (Ψ), respectively. The specific augmentations are conducted using Equations 3.41 and 3.42.

The correction step of the Dual EKF is carried out using Equations 3.38-3.40 and still updates the covariance matrix, Kalman gain, and the augmented state vector upon arrival of a new measurement. Once again, for this system, the controllability matrix is assumed to be linear and does not need to be linearized from a nonlinear system model. Prediction is carried out using the remaining equations shown above (3.36 and 3.37) and utilizes the nonlinear system model and the linearized state-space model. During this step, the estimated kinetic parameters are held constant and are only updated during the correction stage of the algorithm.

In terms of novelty, this work builds upon the existing Dual EKF algorithm by combining it with DO to improve the estimation performance. Both the standard and modified DO approaches can still be applied to the Dual EKF algorithm but require more computational time as the size of the optimization problem increases from 16 to 26 variables, due to the incorporation of the kinetic parameters in matrix Q (Q now has dimensions of 20×20). For brevity, only the modified DO algorithm is applied with the Dual EKF here as it has shown better estimation performance than the standard formulation. Once again, to guarantee feasibility, constraints are placed to bound the covariance values between 0 and 10.

The derived covariances are used with the previous data sets (i.e., the simulated plant data from Section 3.2.2) to produce the estimation results shown in Figures 3.9 and 3.10 and fairly compare the Dual EKF to the traditional EKF algorithm. Figure 3.9 specifically shows the Dual EKF performance in terms of the state variables.

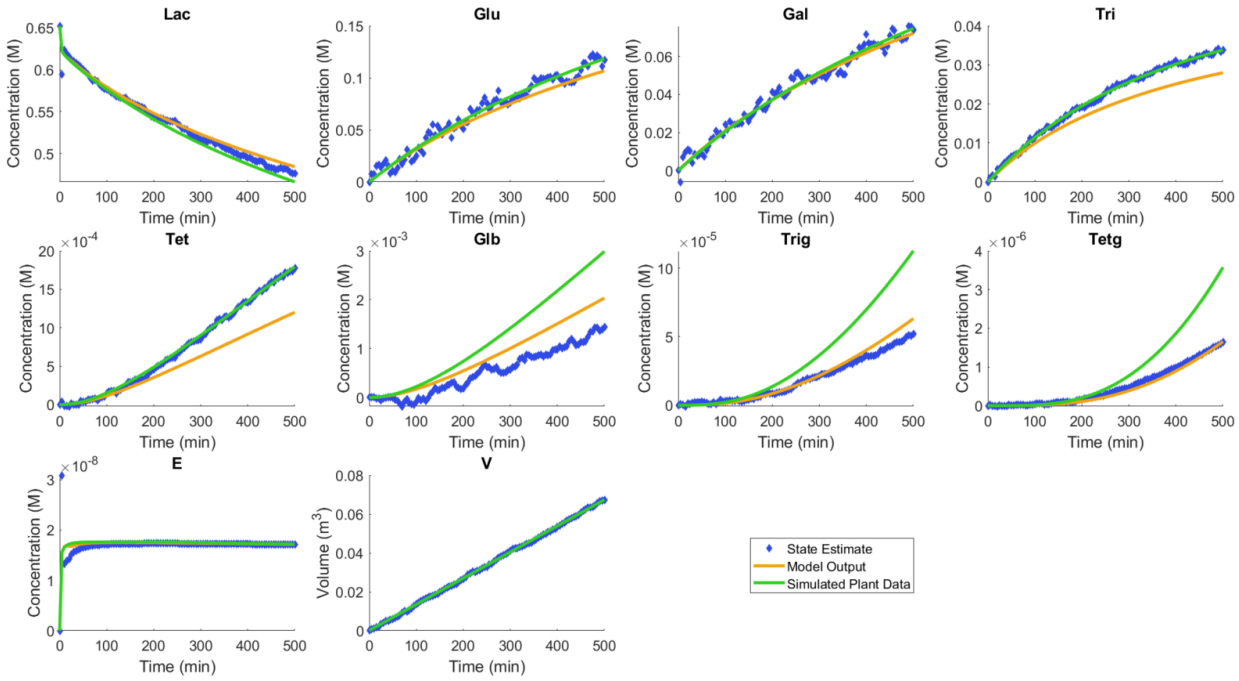

Figure 3.9 Dual EKF estimation performance for state variables using Modified DO

Most of the state estimates shown in Figure 3.9 either lie within or converge to the simulated plant data or model output data sets. States *Glb* and *Trig* fall outside of these bands but closely follow the model output data and trajectory of these curves. This is an improvement from the previous EKF results, where the *Glb* and *Trig* state estimate deviations were more oscillatory in nature and larger in magnitude. State *Gal* still maintains an oscillation around the data sets, but the magnitude of the oscillations is much less severe using the Dual EKF. Overall, in terms of the state variables, the Dual

EKF performs well and generates satisfactory and more realistic approximations of the true state values when compared to the traditional EKF. Figure 3.10 shows the estimation performance in terms of the measurements and demonstrates that the Dual EKF is also able to filter out a large amount of the measurement noises.

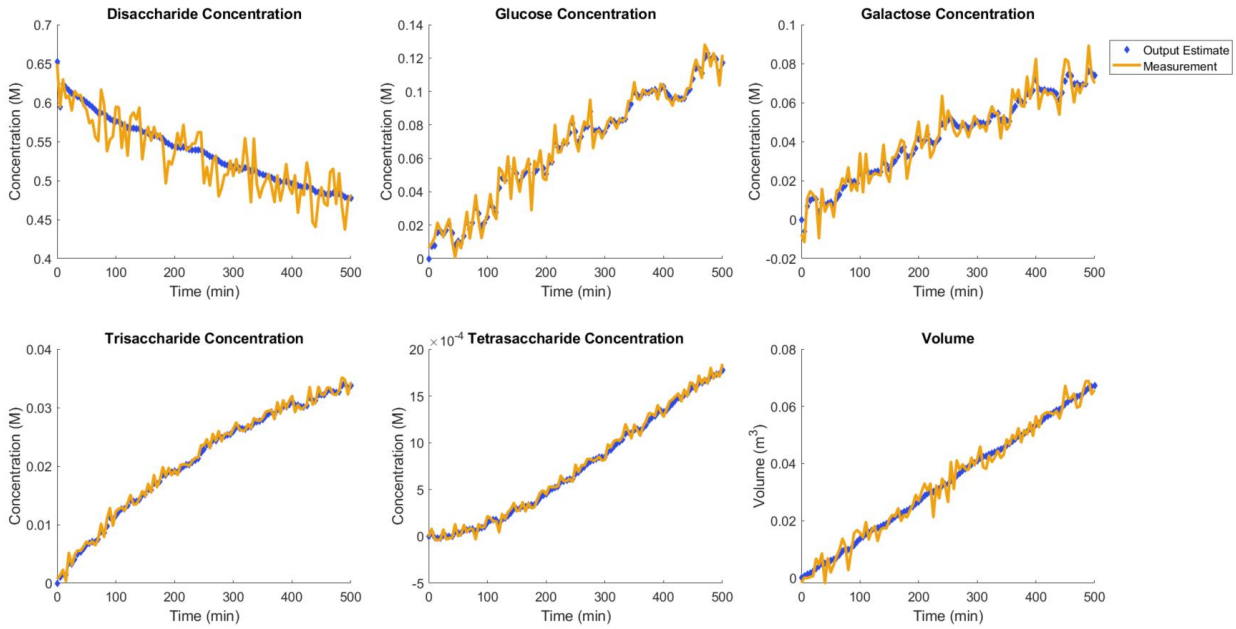

Figure 3.10 Dual EKF estimation performance for outputs using Modified DO

Figure 3.10 demonstrates that the Dual EKF can filter out a large portion of the white noise applied to the measurements as the output estimates generally lie in between the oscillations of the measurements. Using the Dual EKF, the *Galactose* output estimates no longer experience the rapid oscillations found in the previous estimation implementations. The remainder of the output measurements closely follow the previous case study trends and exhibit excellent behavior. Overall, of the EKF-based estimation approaches applied to this system, the Dual EKF is able to derive the most accurate and converged state estimation results.

3.4 Comparison of EKF Algorithms

Although the case studies are effective at highlighting the estimation performance in terms of the state variables, it does not use a quantity to measure the estimation error between the state estimates and the true state values. To quantify the performance of the different EKF techniques, the squared error (SE) between the state estimates and the simulated plant model is used and is defined in Equation 3.43.

$$SE = \sum_{k=1}^{K} (x_{plant\ k} - \hat{x}_k)^2 \tag{3.43}$$

In industrial applications, calculation of the SE is often challenging as the true state values are unknown or require expensive and time intensive procedures to gather. However, in this work the simulated plant model serves as the source of the true state values as these are derived using the simulated plant kinetics. This allows the SE formulation defined above to be used to measure estimator performance. In this work, an SE of 0 implies that the state estimates perfectly reflect the true process state values (i.e., the simulated plant model), thus providing the best possible outcome. As the estimation performance may vary across different data sets, Table 3.4 shows the SE for the case studies discussed above and the average SE from running 100 estimator implementations using randomized values for lactose to enzyme ratios (between 3-25) and total feed rates (between 0-0.001667 m^3/min) entering the reactor.

Table 3.4 SE values for EKF-based estimation

EKF – Standard DO		**EKF – Modified DO**		**Dual EKF – Modified DO**	
Case Study SE	Average SE	Case Study SE	Average SE (*)	Case Study SE	Average SE (*)
0.0549	0.0457	0.0125	0.0128	0.0096	0.0078

The asterisks indicate that one or more simulations were unstable and produced an SE value orders of magnitude higher than the average. These values were neglected for the average SE calculation.

When running the 100 simulations, the same data sets are used for all EKF implementations to fairly compare the various implementations. Overall, the EKF with standard DO had the largest SE values while the Dual EKF had the lowest SE values. Intuitively, these SE results are rational because the EKF with standard DO estimates the covariances for minimizing the measurement error, which leads to significant oscillations from the true states. Furthermore, the standard EKF always relies upon the nominal kinetic parameters for the estimated forecast, rather than parameters being estimated in the Dual EKF. In turn, this produces more forecasting error and contributes to producing higher SE values. Due to these two factors, the EKF with standard DO has approximately 6 times higher SE value than the Dual EKF.

Figure 3.11 breaks down the individual simulation SE values for each EKF algorithm and supplies insight into how robust each algorithm is when applied to this system. Robustness in this dissertation refers to the ability of a state estimation algorithm to produce feasible, but not necessarily optimal estimation results.

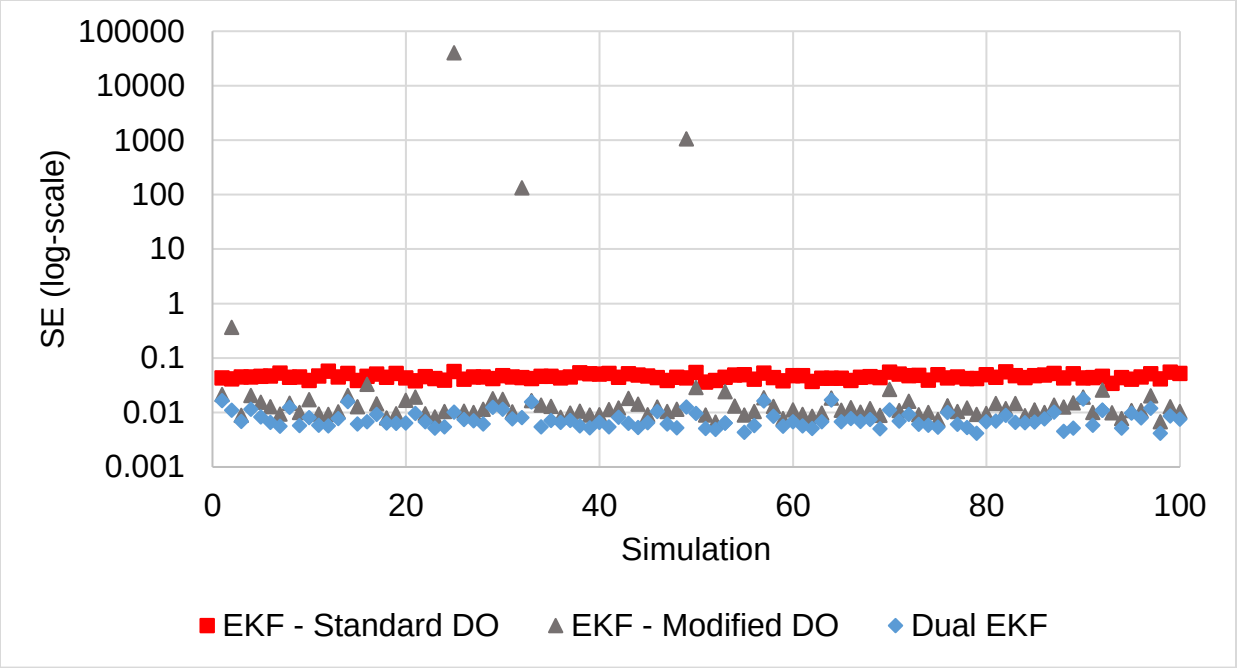

Figure 3.11 SE values for EKF simulations performed

Due to the range of SE values in Figure 3.11, a semi-log plot is used to show the SE values of the different estimation techniques being tested. Robustness is inferred by counting the number of simulations that diverge significantly (i.e., by orders of magnitude) from the average SE values or produce a *NaN* error, with more simulations diverging inferring less robustness. The EKF with standard DO produces a range of SE values between 0.0342 and 0.0574 and has no simulations diverging from this region. The EKF with modified DO produces an SE range around 0.01 and 0.02, with 4 simulations falling far outside this SE range. The Dual EKF produced an SE range between 0.0043 and 0.0178 and has two simulations with an SE of *NaN* (not shown above), due to instability of the estimation algorithm. Based upon these results, the EKF with standard DO tuning exhibits more robustness considering these simulations, as all the simulations produce valid estimation results. Although there is some arbitrary

decision making currently being employed to determine when the simulations diverge, an excellent tool that could be utilized are box-and-whiskey plots. These plots are useful for comparing relative values and can point out exceptionally large or small values with ease, thus determining when outlier simulation results are present (Larsen, 1985).

Although it is difficult to track the root cause of each simulation failure, it appears that the high SE values or *NaN* results are the product of negative state estimates behaving poorly with the model. For example, in simulation 2 of the EKF with modified DO, *Glb* begins to experience negative state estimates beyond discrete time step 3. Over time this drives the *Lac* state estimates to much higher values than predicted by the model. This error continues to rapidly build in these specific state variables, but eventually propagates through the other states, thus leading to poor estimates of all state variables. The propagation of error occurs due to the interrelated nature of all state variables in the model.

Due to the random plant kinetic parameters and joint measurements of states such as *Glb* and *Lac*, it may be possible that certain randomized studies are not well represented by the DO derived covariances. In these cases, the EKF attempts to filter out the measurement noise and does so by overestimating the value of *Lac* and underestimating the value of *Glb,* in some cases to the point where *Glb* is assigned a negative value. Theoretically, it may be possible to apply clipping or another EKF based algorithm discussed in the literature review to prevent negative estimates, but this is outside the scope of this work as it may not be generalizable to other bioprocesses.

As EKF-based estimation can be sensitive to poor initialization conditions, the initial enzyme concentration is now varied from 0 M to 3.63×10^{-7} M and then used to initialize the EKF simulations. By varying the initial conditions, estimation robustness can be examined again considering different startup conditions. Table 3.5 shows the SE values for the different estimation techniques across 100 simulations.

Table 3.5 SE values for EKF-based estimation with modified initial conditions

EKF – Standard DO	**EKF – Modified DO**	**Dual EKF**
Average SE (*)	Average SE (*)	Average SE (*)
0.0496	0.0530	0.0575

The asterisks indicate that one or more simulations were unstable and produced an SE value orders of magnitude higher than the average. These values were neglected for the average SE calculation.

When varying the initial conditions, all EKF implementations produce some number of infeasible sets of state estimates that must be rejected when calculating the average SE values. Overall, there is little difference in the average SE values when directly applying the EKFs with the new set of initial conditions. These average SE values are greater than those from Table 3.5 and could potentially be improved by re-running the covariance estimation with the new initial conditions. Figure 3.12 highlights a significant decrease in the robustness of all the EKF algorithms when using this new set of initial conditions.

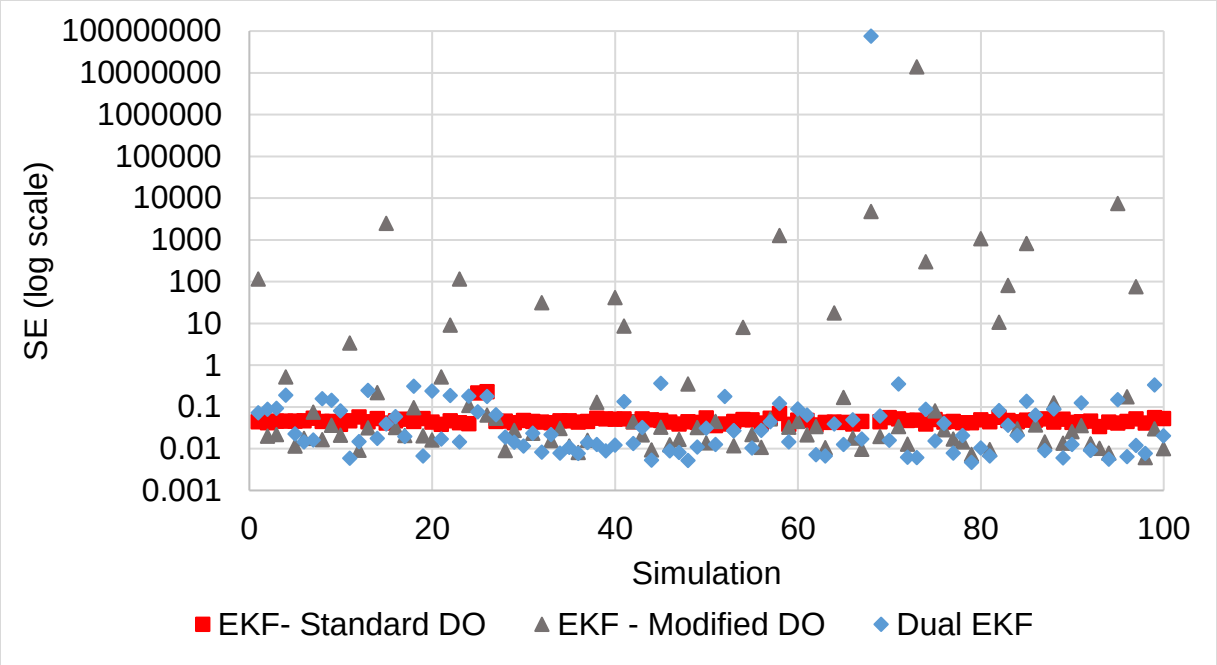

Figure 3.12 SE values for EKF simulations with modified initial conditions

When looking at Figure 3.12 many of the simulations diverge significantly from the low SE values and are very sporadic. In terms of robustness, the EKF with standard DO tuning produced 1 set of invalid state estimates, the EKF with modified DO tuning produced 25 sets of invalid state estimates, and the Dual EKF produced 4 invalid sets of state estimates. Overall, when modifying the initial conditions, all the EKF-based estimation algorithms fail to produce constant and reliable estimation results which could introduce some challenges for implementing this technology into an industrial application. As a result, the initial state estimates must be carefully chosen to help promote feasible estimation results when using EKF-based estimation approaches.

Although EKF-based estimation techniques cannot guarantee feasibility for all scenarios, the recursive nature of the algorithm allows near instant derivation of state estimates, which is critical for online estimation applications. Table 3.6 outlines the

computational time required to run the EKF-based estimation algorithms considering the EKF simulations from Table 3.4.

Table 3.6 Computational times for EKF-based estimation

EKF – Standard DO		**EKF – Modified DO**		**Dual EKF**	
Case Study	100 Sims	Case Study	100 Sims	Case Study	100 Sims
0.32 (s)	6.85 (s)	0.32 (s)	11.23 (s)	0.33 (s)	31.92 (s)

The computational times outlined in Table 3.6 consider running the entire estimation algorithm, which includes all 101 iterations of the EKF for each simulation. Case study time refers to the time required to derive the state estimates associated with the datasets from Figures 3.7-3.10, while the “100 Sims” time refers to the total time required to derive all state estimates for all simulations of Figure 3.11. The DO tuning part of the framework is not included when calculating simulation time as this is conducted offline and is not directly used in the day-to-day process monitoring operations. Overall, all these EKF-based estimation algorithms are able to generate near instantaneous state estimates and take under a minute to run 100 implementations of their algorithm (which generates over 10,000 sets of state estimates). Among these algorithms, the Dual EKF has the highest computational time and is likely caused by the increased dimensionality of the system covariances used to accommodate the estimation of the kinetic parameters. Despite this, all these EKF formulations can be applied to a real process for online estimation of the process states, at the risk of lack of feasibility approximately 2-4% of the time, based off the data in Table 3.4.

3.5 MHE-Based Estimation Approach

After reviewing the failure rates in the previous section, MHE was applied to guarantee feasible state estimates. As noted in Section 2.1, MHE uses an optimization-based approach to derive estimates of process states which directly allows constraints to be incorporated into the estimation problem. For this system, the much higher computational time of MHE is not a significant concern as the process has relatively slow dynamics and does not require rapid intervention. For reference, the canonical formulation of the linear MHE algorithm is shown below as Equation 3.44 (Rawlings et al., 2022).

$$\hat{V}_T\big(x_N(T)\big) = \hat{V}^-_{T-N}(x_{T-N}) + \frac{1}{2}\left[\sum_{K=T-N}^{T} |y_k - C\hat{x}_k|^2_{R^{-1}}\right] + \frac{1}{2}\left[\sum_{K=T-N}^{T-1} |\hat{x}_{k+1} - A\hat{x}_k|^2_{Q^{-1}}\right] \quad (3.44)$$

In Equation 3.44, the term $\hat{V}^-_T(x_{T-N})$ refers to the arrival cost, the middle term incorporates measurements into the algorithm, and the last terms incorporates the process model into the algorithm. The notation $|X|^2_Y$ corresponds to a quadratic error function ($X^T Y^{-1} X$), that is calculated across each point in the horizon and is eventually summed. The horizon length (*N*) is tuned to balance computational time and potential estimation performance as these are generally inversely related. Based upon Alexander et al. (2020), this estimation problem is solved via the *modSQP* algorithm as it has exhibited an order of magnitude reduction in computational over MATLAB's *fmincon* algorithm. Due to this reduction in computational time and slow process dynamics, the MHE algorithm in this work does not need to use a linearized reactor model during the forecasting phase. Instead, the nonlinear model is utilized to eliminate error from linearization, thus reducing associated modeling inaccuracies.

The canonical MHE formulation is adapted to handle nonlinear forecasting of the state estimates by simply modifying Equation 3.44 into the form shown in Equation 3.45. The arrival cost is not included in Equation 3.45 as it was set to a value of zero in the upcoming case study in Section 3.5.1. Additionally, in this implementation, all state variables are constrained to be nonnegative as defined in Equation 3.46.

$$\hat{V}_T\left(x_N(T)\right) = \frac{1}{2}\left[\sum_{K=T-N}^{T} |y_k - C\hat{x}_k|^2_{R^{-1}}\right] + \frac{1}{2}\left[\sum_{K=T-N}^{T-1} |\hat{x}_k - f\hat{x}_k|^2_{Q^{-1}}\right] \tag{3.45}$$

Subject to the following constraint for this process:

$$\hat{x}_k \geq 0 \tag{3.46}$$

To assist in selecting a reasonable horizon length for MHE, a balance between minimizations of the SE and computational time is established through running simulations by varying the horizon length and examining these objectives. The results of this study are shown in Table 3.7.

Table 3.7 MHE case study examining horizon length selection

N	Total SE across 5 simulations	Computational Time (s)	Time per Iteration (s)
1	0.157	131.56	0.26
2	0.132	463.17	0.93
3	0.126	866.34	1.73
4	0.123	1439.88	2.88
5	0.122	3525.09	7.05

The results shown in Table 3.7 are generated using the synthetic measurement framework described earlier, to create 5 randomized data sets that remain constant across all horizon lengths tested. The computational time refers to the entire time

required to run all 5 MHE implementations completely and the time per iteration refers to the average amount of time required to derive a single set of state estimates, while using *modSQP*. For online implementation, the average time per iteration cannot exceed 300 seconds (5 minutes) and should be well below this limit to ensure that state estimates are always available when the control law needs to be updated. Based upon Table 3.7, going above a horizon length of 3 or 4 provides little reduction in the estimator SE error and increases the computational time several folds with the horizon length. As a result, the upcoming MHE case study in Section 3.5.1 uses a horizon length of 4.

3.5.1 MHE Case Study

To compare the estimation performance of the MHE to the EKF, the same data sets introduced above are reused in the MHE case study. Covariance estimation coupled with MHE is significantly more challenging due to the inclusion of linear and nonlinear constraints on state estimates and a higher computational time. Although it is theoretically possible to combine DO and MHE, this would ultimately form a bilevel optimization problem that would be intractable online. As a result, the MHE covariances are estimated in this section by performing extensive simulations until acceptable estimation results are achieved. It is possible to reuse the same covariances from the EKF case study, but this resulted in worse estimation performance for the MHE case study. For the case study shown in Figure 3.13, a diagonal Q matrix (of size 10x10) with value 10, a diagonal R matrix (of size 6x6) with value 1, and a horizon length of 4 are applied.

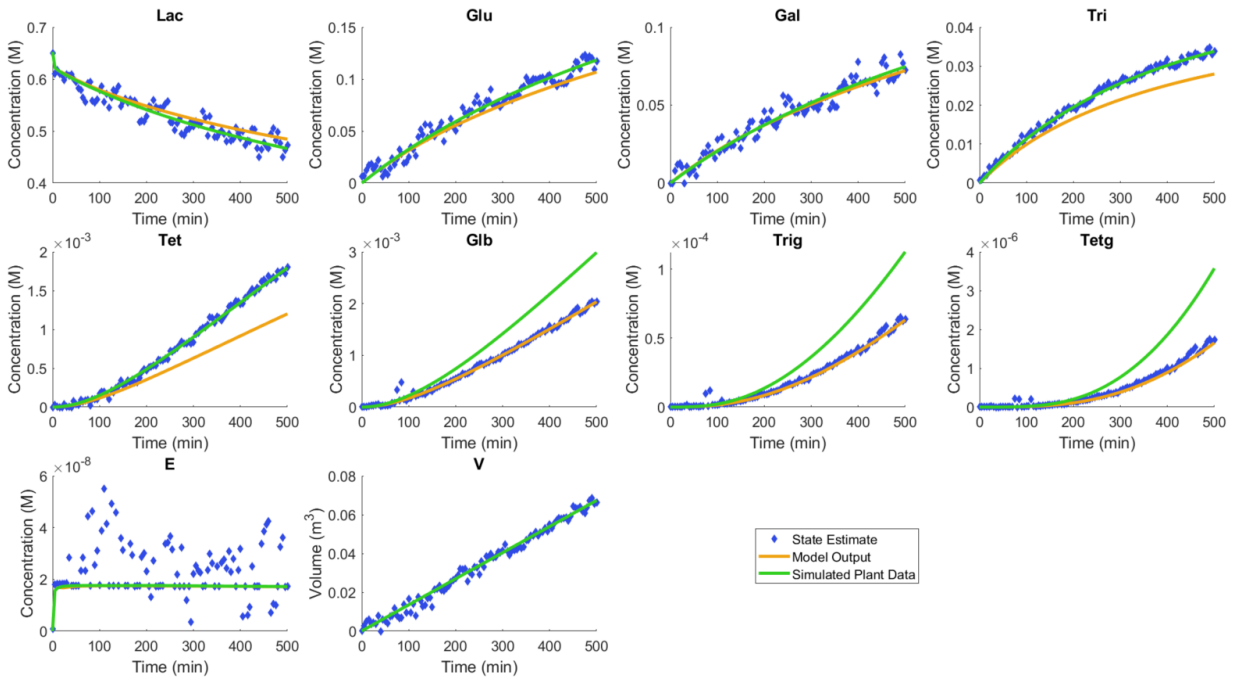

Figure 3.13 MHE performance results for state variables using modified covariances

When using these covariances, minimal offset occurs between most state estimates, the model output and simulated plant data. States *Tri*, *Tet*, *Glu*, and *Volume* converge to the simulated plant data and produce smooth state estimates. States *Glb*, *Trig*, and *Tetg* converge to the model output and generally have smooth state estimates. State *Tetg* experiences some oscillations beneath 100 minutes, but eventually converges and smooths out. Despite having feasible and good estimation performance for many of the state variables, the enzyme concentration (*E*) estimates oscillate significantly from time step to time step and fail to converge to the simulated plant data. This likely occurs as this state variable is entirely derived from the model output, and the MHE algorithm simply fits this state variable to minimize the offset between the other state variables, the model output, and the measurements. Figure 3.14 shows the estimation performance in terms of the output estimates.

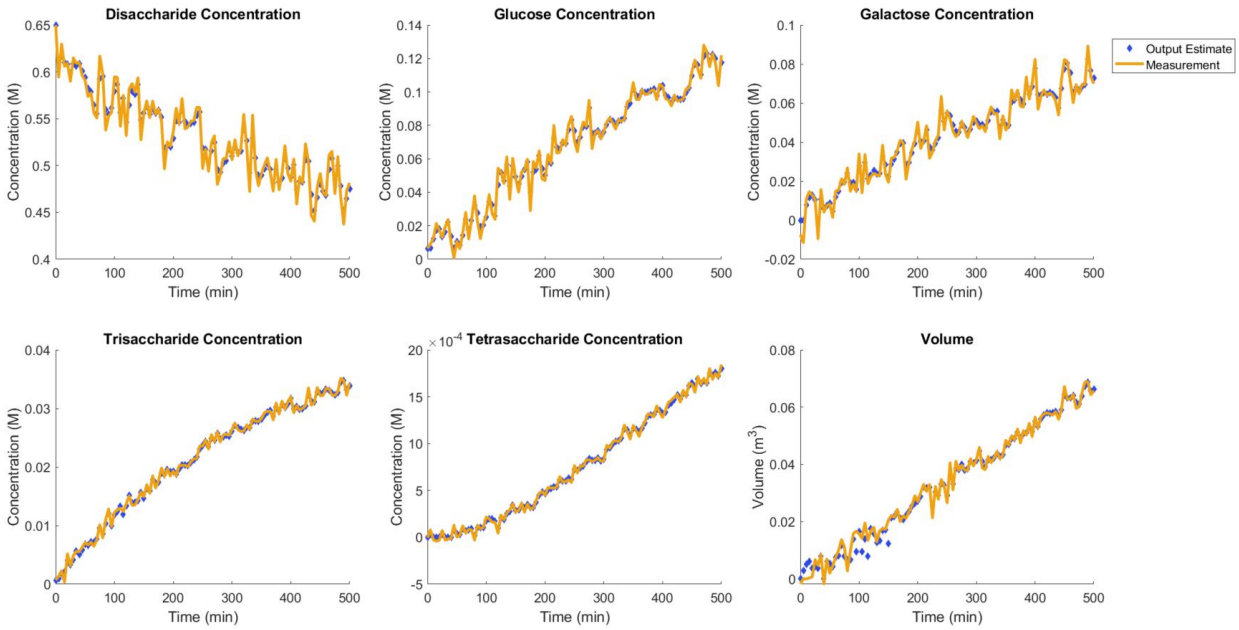

Figure 3.14 MHE performance results for outputs using modified covariances

In terms of the output estimates, the relative weights of the covariances reduce the significance of the measurements, thus minimizing the oscillatory behavior of output estimates. Based upon the results of these case studies, it appears that the traditional MHE algorithm struggles to successfully produce reliable estimates of the enzyme concentration. Although the goal of this work is to estimate and monitor the production of GOS, knowledge of the enzyme concentration may be critical in a real implementation. As a result, the standard MHE algorithm cannot be relied entirely upon for this specific process, despite its ability to guarantee feasible state estimates. To retain this benefit and attempt to improve the enzyme concentration state estimates, a joint parameter and state estimation MHE framework is formulated and implemented next.

3.5.2 Proposed Parameter-Based Moving Horizon Estimation (P-MHE)

Joint parameter and state estimation using MHE approaches have been addressed in the literature using a variety of methods (Kühl et al., 2011) (Sun et al., 2015) (Rodríguez et al., 2021). In this work, a novel approach is applied to estimate the kinetic parameters and the state variables concurrently and is referred to as the Parameter-based Moving Horizon Estimation (P-MHE). This novel P-MHE algorithm is a constrained optimization-based estimation technique that modifies the canonical MHE formulation to directly include the kinetic parameters into the estimation algorithm. The derivation of the P-MHE is outlined in Equations 3.47-3.51.

Standard MHE formulation:

$$\hat{V}_T\big(x_N(T)\big) = \hat{V}^{-}_{T-N}(x_{T-N}) + \frac{1}{2}\left[\sum_{K=T-N}^{T} |y_k - C\hat{x}_k|^2_{R^{-1}}\right] + \frac{1}{2}\left[\sum_{K=T-N}^{T-1} |\hat{x}_{k+1} - A\hat{x}_k|^2_{Q^{-1}}\right] \quad (3.47)$$

For a nonlinear system, apply the following assumptions:

$$\hat{V}^{-}_{T-N}(x_{T-N}) \approx \frac{1}{2}|x_{T-N} - f(\hat{x}_{T-N-1}, u, \rho)|^2_{(Q)^{-1}} \quad (3.48)$$

$$\frac{1}{2}\left[\sum_{K=T-N}^{T} |y_k - C\hat{x}_k|^2_{R^{-1}}\right] \approx \frac{1}{2}\left[\sum_{K=T-N}^{T} |y_k - Cf(\hat{x}_{T-N}, u, \Delta t)|^2_{R^{-1}}\right] \quad (3.49)$$

$$\frac{1}{2}\left[\sum_{K=T-N}^{T-1} |\hat{x}_{k+1} - A\hat{x}_k|^2_{Q^{-1}}\right] = 0; Replace\ with \frac{1}{2}|p_{kin} - \hat{p}_{kin}|^2_{(\theta)^{-1}} \quad (3.50)$$

Resulting in the P-MHE formulation:

$$\hat{V}_T\big(x_N(T)\big) = \frac{1}{2}|\hat{x}_{T-N} - f(\hat{x}_{T-N-1}, u, \rho)|^2_{(Q)^{-1}} + \frac{1}{2}\left[\sum_{K=T-N}^{T} |y_k - Cf(x_{T-N}, u, \Delta t)|^2_{R^{-1}}\right] +$$

$$\mathbf{\frac{1}{2}}|p_{kin} - \hat{p}_{kin}|^{\mathbf{2}}_{\boldsymbol{(\theta)^{-1}}} \quad (3.51)$$

Equation 3.47 is the standard MHE formulation (also seen in Equation 3.43) that does not identify the source of model noise and requires *N* sets of state variables to be incorporated into the optimization problem. As a result, as *N* increases, there is a subsequent increase in the number of decision variables, optimization complexity, and computational time. For this bioprocess application, a number of assumptions can be applied to modify Equation 3.47 and reduce the computational time of the algorithm.

In this particular process application, it is assumed that all model uncertainty and subsequent model error is the result of a difference between the nominal kinetic values and the true plant kinetic values. As a result, these kinetic parameters can be directly embedded into the estimation algorithm for their calculation. Furthermore, if the estimated kinetic parameters are assumed constant for an estimation horizon of *N* points, then any forecasting of the state estimates should have reduced error (as most model error is assumed to come from the kinetics). This ultimately produces the assumptions shown in Equation 3.50 and only requires a single set of state estimates and the kinetic parameters to be fit for an estimation horizon of *N*, thus shrinking the size of the optimization problem. Equation 3.48 defines the arrival cost as the difference between the current state estimates being calculated ($\hat{x}_{T-N}$) and the forecast of the previous state estimates using the nonlinear reactor model ($f(\hat{x}_{T-N-1}, u, \rho)$). In many MHE applications, the arrival cost is defined by the user, so this one stage arrival cost is assumed here. Equation 3.49 uses the state estimate and the nonlinear model to derive the error between the measurements and state estimates.

3.5.3 P-MHE Case Study

Figure 3.15 shows the application of the P-MHE to the same data set used in the other estimation case studies.

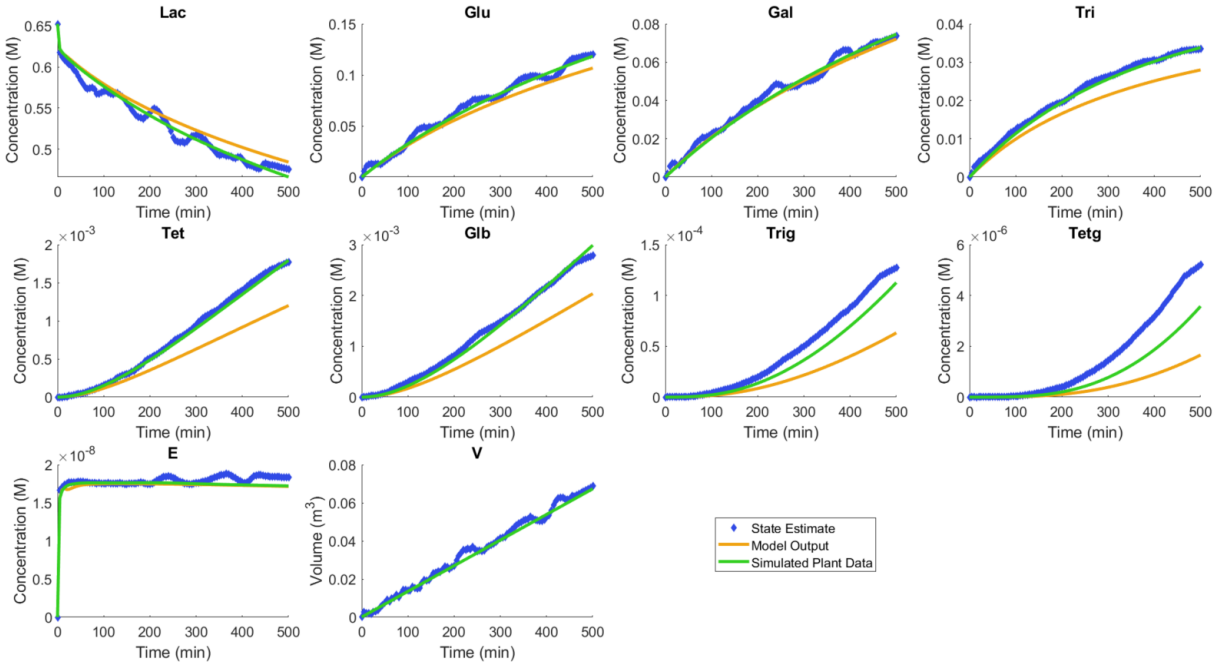

Figure 3.15 P-MHE performance results for state variables

Due to the constrained nature of the P-MHE algorithm, all state estimates generated are guaranteed to be feasible, which can be seen by all states having estimates greater than or equal to 0. In terms of estimation performance, P-MHE produces state estimates that generally lie in-between the simulated plant data and the model output with most state estimates converging to the simulated plant data. Only states *Trig* and *Tetg* lie outside the model output and simulated plant data, but closely follow the trajectories and values of the simulated plant data. Moreover, the first data point for the state estimate of enzyme concentration (*E*) does not strictly follow the model

output and simulated plant data, as a sufficiently long horizon length was chosen (*N*=5) that neglected the very rapid change in enzyme concentration at the start of the batch. This rapid change occurs because the reactor is initialized with no enzyme and a very small amount of lactose (*Lac*). Overall, P-MHE provides accurate and guaranteed feasible state estimates for all states. Figure 3.16 shows the P-MHE estimation performance considering the output measurements.

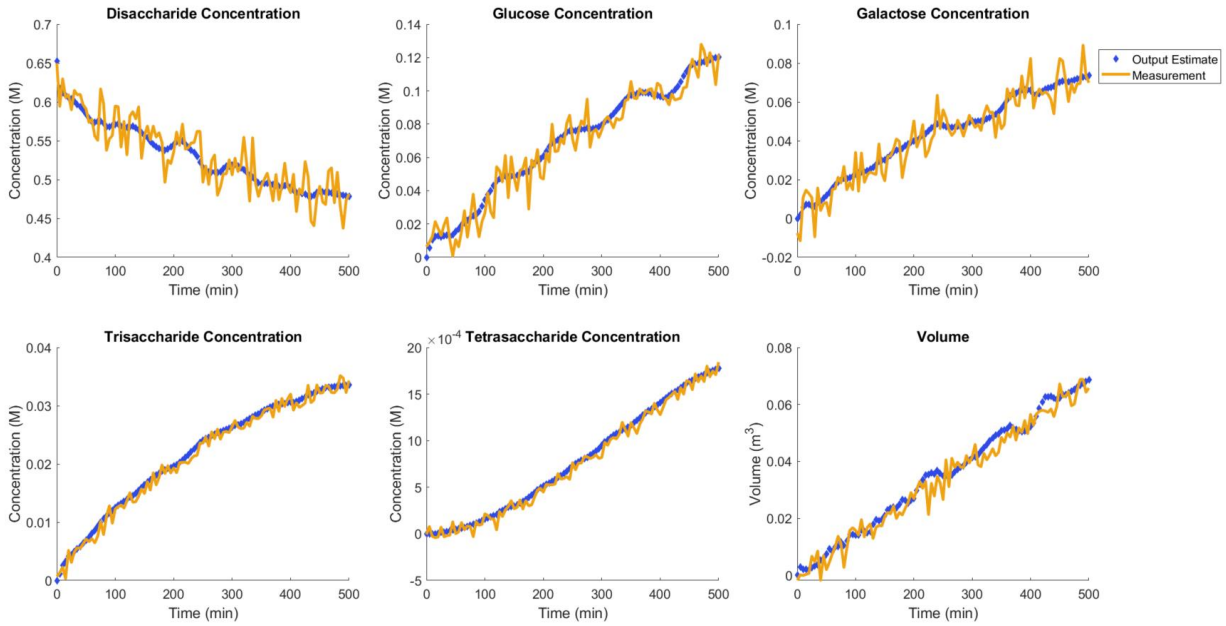

Figure 3.16 P-MHE performance results for output measurements

The P-MHE does an excellent job at reducing the effects of the white noise on the measurements as the output estimates lie within the peaks and valleys of the measurements. Additionally, the output estimates are smooth and free from the rapid oscillations found in the measurements. At times the volume output estimate does deviate from the volume measurement, but generally converges to the measurements. This behavior and the estimation of *Trig* and *Tetg* can likely be remediated in future P-

MHE implementations by further tuning the covariances using another systematic approach from the literature such as ALS. Furthermore, due to the flexibility of this algorithm, additional constraints could be implemented to bound the estimates to certain regions of acceptable values.

3.5.4 Comparison of MHE Algorithms

Due to the use of optimization algorithms in MHE, there is a much greater computational time for this estimator when compared to EKF and other recursive techniques. In this work, the computational time of the MHE is limited by selecting an approximate and reasonable horizon window and selecting the more efficient *modSQP* technique over standard optimization algorithms such as in *fmincon*. Overall, each specific MHE implementation can vary significantly with respect to computational time and estimation accuracy. The results of this implementation in terms of the SE values and computational times are shown in Tables 3.8 and 3.9, respectively.

Table 3.8 SE values for MHE-based estimation

MHE SE		**P-MHE SE**	
Case Study	25 Sims	Case Study	25 Sims
0.0138	0.3057	0.0086	0.278

Table 3.9 Computational times for MHE-based estimation

MHE CPU Time		**P-MHE CPU Time**	
Case Study	25 Sims	Case Study	25 Sims
524.72 (s)	9961 (s)	179.33 (s)	2654 (s)

Due to the 3-4 orders of magnitude increase in computational time of the MHE, it is impractical to run 100 simulations to examine performance. As a result, to examine the feasibility, estimator robustness, and average computational time across multiple iterations, only the first 25 data sets from the EKF simulations are used with the MHE. Using the original MHE, it takes approximately 525 seconds to derive 100 sets of state estimates (≈5.5 seconds/measurement) of the MHE algorithm for all measurements. The P-MHE minimizes the computational time of this class of estimation problem and produces an average computational time of approximately 1.7 seconds/measurement as fewer variables are being optimized concurrently, thus reducing the complexity of the optimization problem.

In terms of robustness, all MHE implementations performed excellently and produced feasible estimation results for all simulations, unlike the EKF-based estimation approaches. In terms of estimation error, the MHE performed marginally worse than the P-MHE. Intuitively, this result makes sense as the original MHE implementation does not have a method of overcoming the chronic plant-model mismatch. The P-MHE restructures the MHE into a joint parameter and state estimation problem which helps to overcome the plant-model mismatch, thus minimizing the SE further than the traditional estimation algorithm. Despite not applying any systematic tuning approach for the P-MHE, further reductions in the SE may be possible with additional tuning. To further test the robustness of this class of estimation algorithm, the modified initial conditions from Table 3.5 are applied to the MHE. The SE and computational time results for the modified initial conditions study are shown in Tables 3.10 and 3.11, respectively.

Table 3.10 SE values for MHE-based estimation with modified initial conditions

MHE SE		**P-MHE SE (using *fmincon*)**	
Case Study	25 sims	Case Study	25 sims
0.0140	0.303	0.0119	0.338

Table 3.11 Computational times for MHE-based estimation with modified initial conditions

MHE CPU Time		**P-MHE CPU Time (using *fmincon*)**	
Case Study	25 sims	Case Study	25 sims
554.16 (s)	10144(s)	237.78 (s)	5601 (s)

For the P-MHE simulations, *fmincon* was used to solve the optimization problem as it did not require any modifications to the algorithm. The current version of *modSQP* can often fail to converge during function evaluations and was particularly prone to failure under these modified initial conditions. Future versions of *modSQP* can hopefully improve upon the stability and robustness of this algorithm. Despite this switch in optimization algorithm, P-MHE still generates a series of state estimates in a relatively short amount of time (every 2.5 seconds/measurement). For the MHE implementation, *modSQP* was still able to be used and did not change the computational times in any meaningful way. Although MATLAB's *fmincon* optimization algorithm is not the most state-of-the-art optimizer available, the results shown in this work can likely be continued to be improved through incorporating more robust optimization techniques. Overall, when using this new set of initial conditions, the MHE-based approaches remained tractable for different initial conditions.

In terms of the estimation robustness, all the MHE-based approaches can adapt to the new initial conditions and consistently produce quality and feasible estimation results. In part this is due to the constraints (i.e., nonnegative concentrations) imposed on the state estimates. Overall, this results in all simulations being included in the total SE calculation as all simulations produced valid state estimates. By modifying the initial conditions, the SE values remain largely consistent for the standard MHE formulations and increase by approximately 22% for the P-MHE. The increase in the SE for the P-MHE places this estimation algorithm on par with traditional MHE techniques, but still has a reduction in the computational time. Overall, the P-MHE algorithm can adapt to the new initial conditions and provide robust estimation performance.

When compared against the EKF-based estimation approaches, the P-MHE algorithm exhibits several benefits. Primarily, the P-MHE algorithm is more robust than the EKF algorithms tested as it guarantees feasibility regardless of the specific initialization conditions due to imposing constraints in the estimation algorithm. Furthermore, the SE of P-MHE is less than or approximately equal to the SE of EKF-based estimation approaches. As a result, the quality of the estimation from the P-MHE is on par with the Dual EKF. Due to the slow process dynamics of this system, estimates are not required in a short amount of time (i.e., seconds). However, for systems where rapid estimates are needed (e.g., safety systems, power systems, rapid reactions) the reduction in computation time achieved with the P-MHE would be significant as it could allow for near-real-time estimates with guaranteed feasibility. Although recursive techniques will almost certainly be faster than any optimization-based method, they cannot guarantee feasibility, thus limiting the usefulness and reliability of these

techniques. Overall factors such as SE value must be considered concurrently with factors such as the computational time and feasibility for any online state estimation implementation. As this P-MHE is a novel algorithm, improvements such as estimating tuning parameters and selecting alternative optimization techniques may continue to improve the estimation results and reduce the computational time.

Overall, the results in this chapter demonstrate the limitations of recursive state estimation approaches for use with this GOS process and the broader semi-batch bioprocess field. The sensitivity to initialization and failure rate of EKF severely limits its ability to be relied upon for quality control applications or feedback control. Based upon these case studies, optimization-based state estimation algorithms such as MHE or P-MHE are highly recommended for real industrial semi-batch bioprocess. These algorithms provide guaranteed estimation robustness and are tractable given they are applied to processes with relatively slow dynamics (which many bioprocesses are).

The main limitation of this chapter was the difficulty in tuning these estimation algorithms. The DO approaches shown in this chapter require offline tuning due to the algorithm being slow and computationally inefficient, even when applied to recursive techniques. For the optimization-based approaches it was simply not tractable to apply this technique. As a result, ad-hoc tuning of these optimization-based algorithms was performed and carries no claim of optimality. These limitations motivate the covariance estimation work highlighted next in Chapter 4 of this dissertation.

Chapter 4 Improved Covariance Estimation Techniques

Building off Chapter 3, the primary aim of this chapter is to improve covariance estimation with DO by employing auto-covariance least squares (ALS). As part of this chapter, the mathematics behind ALS are presented along with a brief explanation outlining what parts of the GNU Octave code had to be updated for the algorithm to run in MATLAB. The MATLAB specific ALS code is examined using a performance assessment with a linear time-invariant (LTI) system and a nonlinear reactor example. Upon this assessment, ALS is applied to the GOS process to improve the state estimation results. As part of this implementation, ALS is directly compared against DO by examining the SE of the associated state estimates and the computational time of both algorithms. Novelty is introduced in several ways including applying and testing ALS to a nonlinear bioprocess system operating in batch/semi-batch mode while undergoing a chronic plant-model mismatch. Ultimately this results in the production of a set of re-deployable ALS codes for other PSE implementations.

4.1 Autocovariance Least-Squares Background

As mentioned in Chapter 2, ALS is a widely used correlation-based covariance estimation technique employing historical measurement data that has demonstrated success with nonlinear systems, batch processes, and semi-batch processes (specific to polymerization applications). The main components of ALS can trace their roots to Odelson et al. (2005), but the mathematical framework presented here is reproduced from Odelson et al. (2005) and Lima et al. (2013). Over time, the specific ALS nomenclature has varied from publication to publication, and the nomenclature

presented in this dissertation makes some modifications to help alleviate confusion and increase consistency between the nomenclature from other chapters.

The original ALS formulation was developed for a LTI, discrete-time state-space system in the form of Equations 4.1 and 4.2, with the assumption that the model and measurement covariances are time invariant in nature.

$$x_{k+1} = Ax_k + Bu_k + Gw_k \tag{4.1}$$

$$y_k = Cx_k + v_k \tag{4.2}$$

In particular, Equation 4.1 propagates the current state variables at discrete time *k* ahead one time step using the control moves (*u*), input matrix (*B*), and linear system matrix (*A*), while accounting for a random white noise using the system noise matrix (*G*) and process model noise (*w*). Equation 4.2 translates the state variables to the corresponding measurements through the linear controllability matrix and random measurement noise (*v*). Over time, this LTI restriction was relaxed and allowed ALS to be applied to the nonlinear stochastic discrete time system shown in Equations 4.3 and 4.4.

$$x_{k+1} = f(x_k, u_k) + G(x_k)w_k \tag{4.3}$$

$$y_k = h(x_k) + v_k \tag{4.4}$$

Much like EKF, ALS cannot directly use the nonlinear stochastic system shown above and instead uses linearizations at different time steps to convert the nonlinear system into the required linear state space form. Ultimately, this linearization of the

nonlinear system produces the set of time-varying equations shown below in Equations 4.5-4.9.

$$x_{k+1} = \bar{A}_k x_k + \bar{B}_k u_k + \bar{G}_k w_k \tag{4.5}$$

$$y_k = \bar{C} x_k + v_k \tag{4.6}$$

$$\bar{A}_k = \frac{\partial f(x_k, u_k)}{\partial x}_{(\hat{x}_{k|k}, u_k)} \tag{4.7}$$

$$\bar{C}_k = \frac{\partial h(x_k)}{\partial x}_{(\hat{x}_{k|k})} \tag{4.8}$$

$$\bar{G}_k = G(\hat{x}_{k|k}, u_k) \tag{4.9}$$

Equations 4.5 and 4.6 are similar to the LTI system but substitute the time-invariant linear matrices (*A*, *C, G*) by time-varying matrices ($\bar{A}_k$, $\bar{C}_k$, $\bar{G}_k$) derived using the linearization Equations 4.7-4.9. This set of linearized equations is used in conjunction with a time-varying state estimator (e.g., EKF) to derive an initial set of state estimates to calculate the state estimation error (ε_k) defined in Equation 4.10 which is evolved (ε_{k+1}) using Equations 4.11-4.14. It is important to note that an initial approximation of the model and measurement noise covariance matrices must be provided to the time-varying estimator to derive this initial set of state estimates.

$$\varepsilon_k = x_k + \hat{x}_{k|k-1} \tag{4.10}$$

$$\varepsilon_{k+1} = \varepsilon_k + \tilde{G}_k \begin{bmatrix} w_k \\ v_k \end{bmatrix} \tag{4.11}$$

$$\tilde{A}_k = (\bar{A}_k - \bar{A}_k L_k \bar{C}_k) \tag{4.12}$$

$$\tilde{G}_k = [\bar{G}_k \quad -\bar{A}_k L] \tag{4.13}$$

$$\breve{w}_k = \begin{bmatrix} w_k \\ v_k \end{bmatrix} \tag{4.14}$$

In the above set of equations, variables $\tilde{A}_k$, $\tilde{G}$, and $\breve{w}_k$ are simply used in the ALS mathematical manipulations and are derived from the linearized time-varying matrices, noises, and Kalman gain. These estimation error equations are used to obtain an alternative state-space model in terms of the L-innovations (i.e., estimator gain) which are employed to define the innovations sequence ($\mathbb{y}_k$) in Equation 4.15.

$$\mathbb{y}_k = y_k - \bar{C}_k \hat{x}_k = y_k - \hat{y}_k \tag{4.15}$$

From here, several assumptions are applied to assist in formulating the ALS problem. The original linear ALS formulation requires three key assumptions to be valid for the process it is being applied to:

1. System *(A, C)* is detectable
2. $\tilde{A}$ is stable: $\prod_m^{m+k-1} \tilde{A} \approx 0$
3. Steady-state initial conditions or selection of k sufficiently large so that initial condition effects can be neglected: $E(\varepsilon_0) = 0,\ cov(\varepsilon_0) = P^-$

The nonlinear form of ALS builds upon this assumption list and requires one additional assumption to be valid.

4. Noise from error due to linearizing the nonlinear model is indistinguishable from other sources of noise and the derived state estimates are considered to be accurate for the employed state estimation technique (Lima et al., 2013).

Equation 4.11 and Assumption 2 allow the innovations sequence to be defined in terms of the linearized controllability matrix, linearized system matrix, linearized

measurement covariance matrix, model noise, and measurement noise as shown below in Equations 4.16 and 4.17. These equations ultimately lead to the formulation of the autocovariance matrix (ACM) as seen in Equation 4.18, where N_{ALS} is the number of lags defined by the user. This lag can be thought of as the ALS equivalent of the MHE horizon length, as a higher number of lags increases the amount of data being used, possibly improving the quality of the covariance estimation results at the cost of increased computational time. In these equations the symbol "$\oplus$" is the direct sum. Equations 4.19 and 4.20 define some mathematical expressions used throughout the ALS formulation and specifically feed into Equations 4.16 and 4.17. Odelson et al. (2005) contains a LTI form of these equations and thus a more convenient expression for the ACM matrix and can be referenced as needed.

$$E[\mathbb{y}_k\mathbb{y}_k^T] = C_k\left[\tilde{A}_{[k-1,1]}\tilde{G}_0 \dots \tilde{G}_{k-1}\right]x \bigoplus_{i=1}^{k} \mathbb{Q}[\tilde{A}_{[k-1,1]}\tilde{G}_0 \dots \tilde{G}_{k-1}]^T C_k^T + R \quad (4.16)$$

$$E\left[\mathbb{y}_{k+j}\mathbb{y}_k^T\right] = C_{k+j}\left[\tilde{A}_{[k+j-1,1]}\tilde{G}_0 \dots \tilde{A}_{[k+j-1,k]}\tilde{G}_{k-1}\right]x \bigoplus_{i=1}^{k} \mathbb{Q}[\tilde{A}_{[k-1,1]}\tilde{G}_0 \dots \tilde{G}_{k-1}]^T C_k^T -$$

$$C_{k+j}\left(\tilde{A}_{[k+j-1,k+1]}\right)A_k L_k R \quad (4.17)$$

$$\mathcal{R}_k(N_{ALS}) = E\begin{bmatrix} \mathbb{y}_k\mathbb{y}_k^T \\ \vdots \\ \mathbb{y}_{k+N-1}\mathbb{y}_k^T \end{bmatrix} \quad (4.18)$$

$$\tilde{A}_{[i,j]} = \prod_{j=k}^{i} \tilde{A}_k = \tilde{A}_i\tilde{A}_{i-1} \dots \tilde{A}_{j+1}\tilde{A}_j \quad (4.19)$$

$$\mathbb{Q} = \begin{bmatrix} Q & 0 \\ 0 & R \end{bmatrix} \quad (4.20)$$

Upon derivation of the ACM, column-wise stacking of the matrix is done to convert the ACM into a form suitable for a least-squares formulation. This column-wise stacking is shown in Equations 4.21-4.26 where "⊗" is the Kronecker product. Equation 4.21 is the main equation and is broken down into smaller expressions (Equations 4.22-4.26) for a more convenient presentation. Variables Γ, Ω_1, Ω_2, Γ_1', Ψ_1 (seen in Equations 4.22-4.26) are ultimately derived from the linearized state space model, the Kalman gains, and initial assumed process model and measurement noise covariances.

$$[\mathcal{R}(N_{ALS})]_S = (\Gamma_1\Omega_1 \otimes \Gamma\Omega_1)\zeta_{g,k}(Q)_s + [(\Gamma_1\Omega_2 \otimes \Gamma\Omega_2)\zeta_{p,k} + I_p \otimes \Psi_1](R)_s \quad (4.21)$$

$$\Gamma = \begin{bmatrix} C_k\left(\tilde{A}_{[k-1,1]}\right) & \cdots & C_k \\ \vdots & \ddots & \vdots \\ C_{k+N-1}\left(\tilde{A}_{[k+N-2,1]}\right) & \cdots & C_{k+N-1}\left(\tilde{A}_{[k+N-2,k]}\right) \end{bmatrix} \quad (4.22)$$

$$\Omega_1 = \begin{bmatrix} G_0 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & G_{k-1} \end{bmatrix} \quad (4.23)$$

$$\Omega_2 = \begin{bmatrix} -A_0L_0 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & -A_{k-1}L_{k-1} \end{bmatrix} \quad (4.24)$$

$$\Gamma_1' = \begin{bmatrix} \left(\tilde{A}_{[k-1,1]}\right)' C_k' \\ \vdots \\ C_k' \end{bmatrix} \quad (4.25)$$

$$\Psi_1 = \begin{bmatrix} I_p \\ -C_{k+1}A_kL_k \\ \vdots \\ -C_{k+N-1}\left(\tilde{A}_{[k+N-2,k+1]}\right)A_kL_k \end{bmatrix} \quad (4.26)$$

In which the symbol $\varsigma_{p,N}$ represents the permutation matrix and converts the direct sum to a vector as defined in Equation 4.27. More details on the mathematical derivation of the ACM and the column-wise stacking process can be found in Lima et al. (2013).

$$\left(\bigoplus_{i=1}^{N} R_v \right)_s = \varsigma_{p,N} (R_v)_s \tag{4.27}$$

These previous equations build upon each other to form the ALS expression presented in Equations 4.28-4.32.

$$\hat{x} = \min_{Q,R} \left\| \begin{bmatrix} \mathcal{A}_k \\ \vdots \\ \mathcal{A}_{N_d-N+1} \end{bmatrix} \begin{bmatrix} (Q)_s \\ (R)_s \end{bmatrix} - \begin{bmatrix} \hat{b}_k \\ \vdots \\ \hat{b}_{N_d-N+1} \end{bmatrix} \right\|_{W_f}^2 \tag{4.28}$$

$$s.t. Q, R \geq 0, Q = Q', R = R'$$

$$\mathcal{A}_k = [\mathcal{A}_{k1} \quad \mathcal{A}_{k2}] \tag{4.29}$$

$$\mathcal{A}_{k1} = (\Gamma_1 \Omega_1 \otimes \Gamma \Omega_1) \varsigma_{g,k} \tag{4.30}$$

$$\mathcal{A}_{k2} = \left[(\Gamma_1 \Omega_2 \otimes \Gamma \Omega_2) \varsigma_{p,k} + I_p \otimes \Psi_1 \right] \tag{4.31}$$

$$\hat{b}_k = \left[\hat{\mathcal{R}}_k (N_{ALS}) \right]_s = \begin{bmatrix} \mathbb{y}_k \mathbb{y}_k^T \\ \vdots \\ \mathbb{y}_{k+N-1} \mathbb{y}_k^T \end{bmatrix}_s \tag{4.32}$$

Equation 4.28 is the final ALS form and is formulated as a positive semidefinite constrained least-squares optimization problem, where the time invariant *Q* and *R* matrices are the decision variables used to minimize the objective function. Unlike DO, ALS does not require state estimates to be rerun as the *Q* and *R* matrices are varied

while minimizing the least-squares objective function, thus significantly reducing the computational time. A schematic for this ALS approach is shown below in Figure 4.1

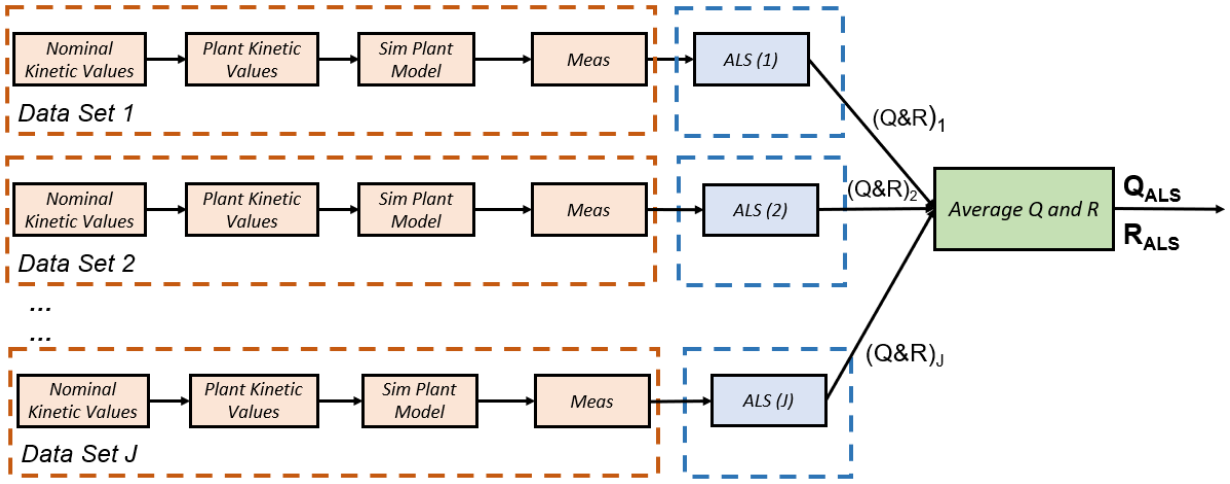

Figure 4.1 ALS framework using simulated data from several simulations

For each data set shown in Figure 4.1, ALS derives a unique set of covariances for each data set tested (as seen with $(Q\&R)_1$, …, $(Q\&R)_J$) and then averages these data set specific covariances to produce the final model and measurement covariance matrices (Q_{ALS} and R_{ALS}). Overall, the setup of ALS requires substantially more effort than DO but uses a more fundamental approach for deriving the covariance estimates based on statistics rather than the more "trial and error" approach of DO.

A key challenge that has not been addressed in the ALS literature is how to define or select an appropriate value for N_{ALS}. Unlike MHE where a heuristic for horizon length has been established (twice as large as system size) (Rao & Rawlings, 2004; Rincon, 2014 B), none has been established for ALS. Much of the ALS literature has cited Lima and Rawlings (2011) which used a N_{ALS} of 15 as it was large enough to provide good covariance estimates. The original ALS publication from Odelson (2005) did not provide

any specific guidance for this parameter and states it is a “user-defined parameter” and the amount of data (N_d) should be far greater than N_{ALS}. Other publications have used alternative lag sizes, but it is well documented that the computational time of ALS is highly dependent on the number of lags. As no pre-established procedure or heuristic has been developed for ALS, this work will choose sufficiently large N_{ALS} values to derive good covariance estimates. Additionally, in this work, it is assumed that larger N_{ALS} values are going to be required for systems of higher nonlinearity or higher dimensionality.

4.2 ALS Code Modification

As noted in the literature review, ALS is currently only available as a series of GNU Octave files which require downloading the Sundials toolbox and linking it to Octave (Rawlings, 2019). For program consistency and more widespread applicability to the general PSE community, a MATLAB specific version of ALS (using mostly standard MATLAB functions) is developed and validated as part of this work. Portions of the GNU Octave codes are directly interchangeable with MATLAB as both languages are high-level programming languages with extensive linear algebra tools. Specifically, most of the basic linear algebra functions are directly interchangeable while functions from MATLAB toolboxes such as the Control System Toolbox and ODE solvers require using alternative Octave functions. Below are some of the changes that were made to develop a MATLAB specific version that will be available as a set of downloadable files, easily deployable to other problems. These codes are being developed to be well documented, user friendly, and serve as a jumping off point for other ALS applications. The final

product will resemble something similar to the *modSQP* MATLAB toolbox (He and Lima, 2020) or the Biologically Inspired Optimal Control Strategy (BIO-CS) MATLAB toolbox (Mirlekar et al., 2017).

1. Octave contains a function for generating the duplication matrix (*duplication_matrix*). MATLAB lacks this function natively, so the MATLAB Central community was consulted for an equivalent function (Jan, 2021).
2. Complete removal of the ODE solvers from the Sundials toolbox (e.g., *CVODES*) and replacement with standard MATLAB ODE solvers (e.g., *ode45* or *ode15*).
3. Removal of structure arrays (e.g., m.x, m.y) for the models and replacement with standard variables.
4. Conversion from the logarithmic barrier function form for the objective function to a standard constrained least-squares estimation problem.

The duplication matrix function is required to generate the final ALS least-squares form and is used throughout the Octave codes. As a result, the equivalent MATLAB function proposed by the MATLAB Central community (self-titled *duplication_matrix.m* as no name was provided) was adapted and validated. For semi-batch processes, linearization is performed using analytical derivatives provided by the user or through the MATLAB symbolics toolbox for difficult to calculate derivatives. At each discretization time the linearization is reperformed. Additionally, some tuning of the N_{ALS} parameter is required to improve the quality of the covariance estimates.

The other significant code change made was the removal of the logarithmic barrier function. ALS was originally posed using a least-squares formulation with an alternative form utilizing a logarithmic barrier function to constrain *Q* and *R* to values greater than zero (Odelson et al., 2005). Although this is a valid form and may be beneficial depending on the optimizer selected, subsequent publications have presented ALS without this logarithmic barrier function (Rajamani et al., 2007; Rincon et al., 2014A; Lima and Rawlings, 2011). As a result, in this work, the logarithmic barrier function was not included as MATLAB optimization algorithms were shown to be generally robust when solving constrained least-squares optimization problems as demonstrated with the MHE results from Chapter 3.

4.2.1 ALS LTI Test Case

To test this MATLAB specific ALS implementation, the LTI system from Odelson et al. (2005), depicted in Equations 4.33-4.39, is used as a case study

$$A = \begin{bmatrix} 0.1 & 0 & 0.1 \\ 0 & 0.2 & 0 \\ 0 & 0 & 0.3 \end{bmatrix} \tag{4.33}$$

$$C = \begin{bmatrix} 0.1 & 0.2 & 0 \end{bmatrix} \tag{4.34}$$

$$G = \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix} \tag{4.35}$$

$$Q_{real} = 0.5 \tag{4.36}$$

$$R_{real} = 0.1 \tag{4.37}$$

$$Q_{guess} = 0.2 \tag{4.38}$$

$$R_{guess} = 0.4 \tag{4.39}$$

In this LTI system, Q_{real} and R_{real} are the true model and measurement covariances that are used to introduce noise to the system, while Q_{guess} and R_{guess} are the initial covariances used to initialize ALS. A KF is used in this LTI example to supply the Kalman gains and state estimates required for ALS. To maintain consistency with Odelson et al. (2005), 200 simulations with 1000 data points and an N_{ALS} of 15 are employed. Figure 4.2 shows the result of this ALS case study.

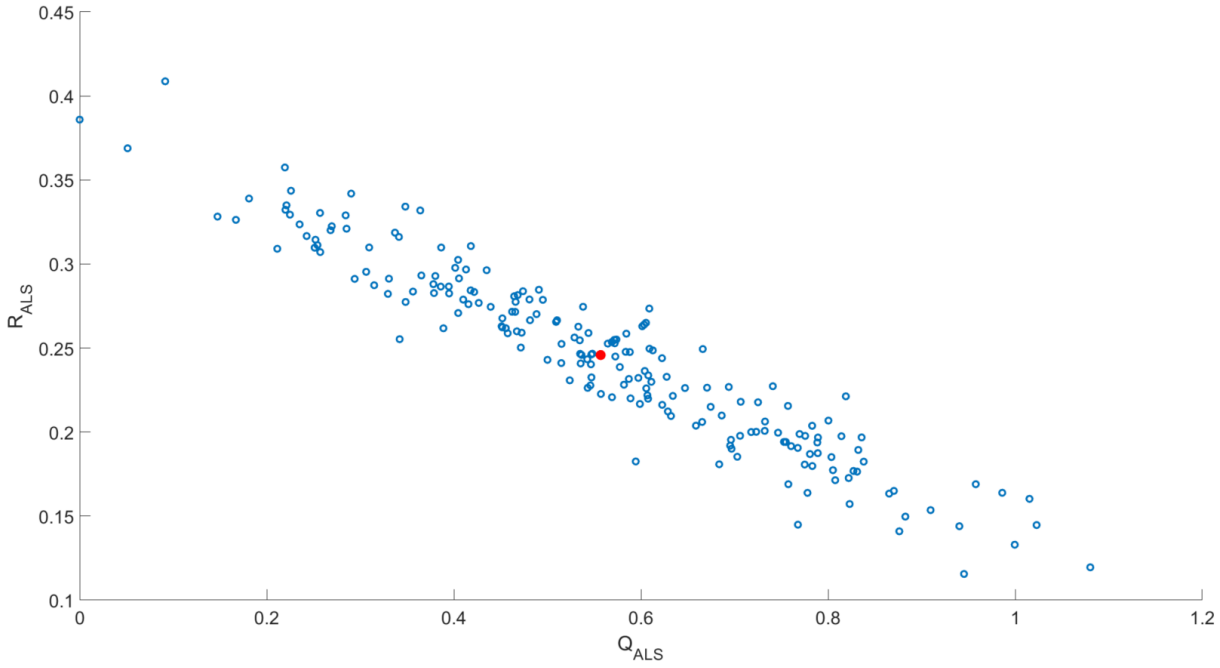

Figure 4.2 ALS results for LTI case study

The blue scatter dots in Figure 4.2 are individual optimal Q and R values derived for each randomized simulation, while the red scatter dot is the average of all simulation results and has a Q_{ALS} value of 0.5559 and an R_{ALS} value of 0.2458. This differs from the results from Odelson as that implementation more closely approached Q_{real} and R_{real}

(many simulations centered around R_{ALS} value of 0.1 and Q_{ALS} value of 0.5) with an Q_{ALS} range of 0.1-0.8 and a R_{ALS} range of 0.02-0.18. One possibility behind this disparity is the difference in ALS formulation as Odelson used a logarithmic barrier function to prevent constraint violation and an undefined optimization technique. In contrast, the MATLAB version simply used the least-squares formulation from Equation 4.28 with the *fmincon* solver with a min-max value constraint of 0 and 2 respectively. Despite not converging as closely to the true covariances, when the derived Q_{ALS} and R_{ALS} are applied to this LTI system using a KF, the SE is minimized as seen in Table 4.1.

Table 4.1 Performance assessment of ALS algorithm using LTI system

Algorithm	Total SE	SE – A	SE – B	SE – C	Computational Time (s)
ALS	16.77	1.27	4.65	10.84	243.93
Real Cov.	19.42	1.47	5.39	12.56	-
Modified DO	26.58	2.05	7.27	17.25	37.73
Standard DO	44.72	3.27	12.73	28.72	36.86
Guess Cov.	21.69	1.66	5.98	14.05	-

To quantity the estimation performance, SE calculations are carried out for states A, B, C, and are summed up to form the total SE value. Within Table 4.1, the Modified DO corresponds to using Equation 3.33 and the Standard DO corresponds to Equation 3.32 to derive the covariances used for the KF. Additionally, the guess covariances use the Q_{guess} and R_{guess} values, while the real covariances employ Q_{real} and R_{real}. Overall, this simple LTI example demonstrates that ALS can improve state estimation results by deriving relatively accurate approximations of the model and measurement covariance matrices. Interestingly, even when using the true system covariances in the KF, the ALS

derived covariances managed to result in more accurate state estimates as seen with the lower SE values. Due to the simplicity of this system, DO is a more computational efficient algorithm as seen with computational times required to derive the estimated covariances. However, real chemical processes are rarely linear in nature so this computational time result cannot be exclusively relied upon.

4.2.2 Nonlinear ALS Test Case

To test the effectiveness of ALS on nonlinear systems, a simple nonlinear case study is carried out using the reaction model shown below in Equations 4.40-4.43 from Alexander et al. (2020).

$$A \underset{k_{-1}}{\overset{k_1}{\rightleftarrows}} 2B \tag{4.40}$$

$$B \overset{k_2}{\rightarrow} C \tag{4.41}$$

$$\frac{dC_a}{dt} = -k_1 C_a + k_{-1} C_b^2 \tag{4.42}$$

$$\frac{dC_b}{dt} = 2k_1 C_a - 2k_{-1} C_b^2 - k_2 C_b \tag{4.43}$$

This reaction model considers the reversible reaction of component A into component B and the conversion of component B into component C using a batch reactor. Of the process states, only component B is measured. In the model above, there are three kinetic parameters of interest, k_1, k_{-1}, and k_2, which are fixed to a value of 1 with appropriate units. Noise is introduced into the system by applying randomized white noises to the process model and the measurements by using the model covariances, measurement covariances, and system noise matrix shown below in Equations 4.44,

4.45 and 4.46, respectively. Random white noise is applied at every discretization point of 0.05 arbitrary time units ranging from 0 to 25. Additionally, the corresponding linearized-state space model, measurement matrix, and initial conditions are shown below in Equations 4.47, 4.48 and 4.49, respectively. As component C does not influence the estimation implementation, it has been neglected from the state-space model for this analysis.

$$Q_{real} = \begin{bmatrix} 1 \cdot 10^{-7} & 0 \\ 0 & 1 \cdot 10^{-7} \end{bmatrix} \tag{4.44}$$

$$R_{real} = [1 \cdot 10^{-4}] \tag{4.45}$$

$$G = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \tag{4.46}$$

$$A_k = \begin{bmatrix} -k_1 & 2k_{-1}C_b \\ 2k_1 & -k_2 - 4k_{-1}C_b \end{bmatrix}_k \tag{4.47}$$

$$C_k = [0 \quad 1] \tag{4.48}$$

$$x_{int} = \begin{bmatrix} 1 \\ 1 \end{bmatrix} \tag{4.49}$$

Both the modified DO and ALS algorithms are applied to the model shown above and employ *fmincon* to derive the covariance estimates. Additionally, 200 randomized simulations are carried out to generate the required plant data and measurement data. In this nonlinear batch process, a N_{ALS} of 25 is utilized for the ALS technique. A sample of the estimation results achieved using the ALS derived covariances combined with an EKF implementation are shown below in Figure 4.3.

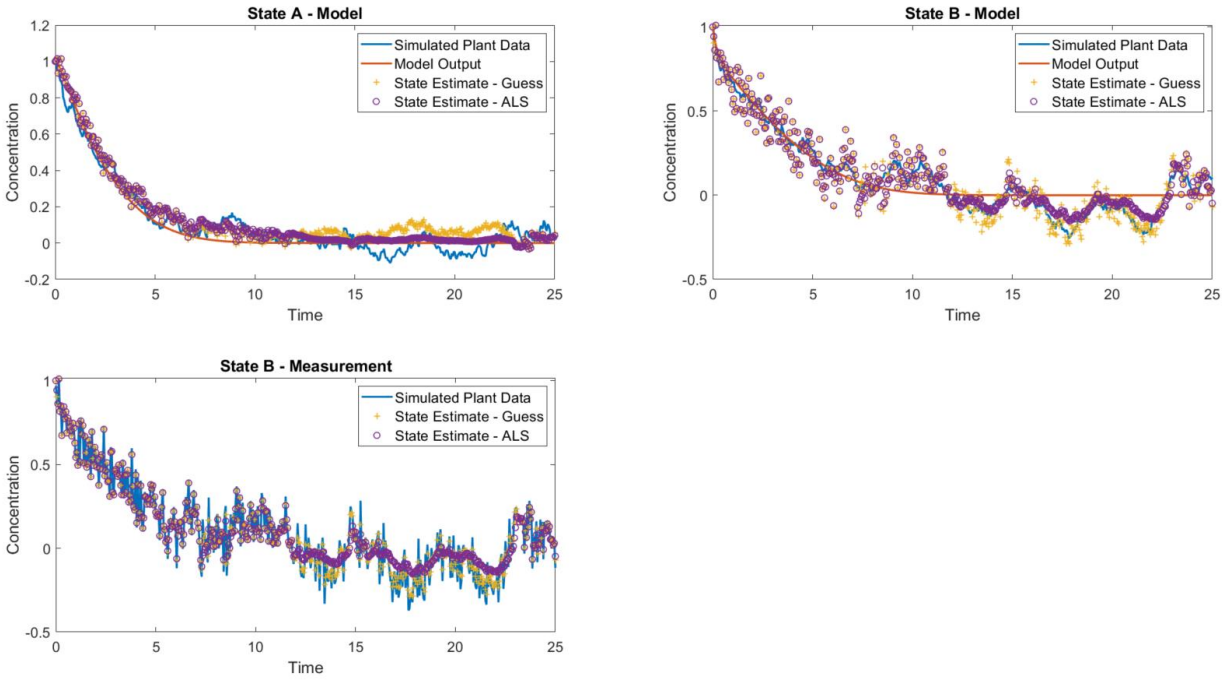

Figure 4.3 Sample of ALS derived covariances for nonlinear EKF example

The ALS state estimates shown in Figure 4.3 use the Q_{ALS} and R_{ALS} derived from the ALS function, while the guess state estimates are derived using the initial guess of the model and measurement covariance matrices. Like in Chapter 3, the simulated plant data represents the true state values, and the model output is the model prediction from the reactor model. Overall, the ALS associated state estimates filter out a large portion of the measurement noise and achieve better estimation results than simply using the initial covariances guesses, particularly for state B. The DO associated state estimation results are not depicted in Figure 4.3 but are reflected in Table 4.2 which quantifies the SE across 200 randomized simulations and the associated computational time to derive the covariances.

Table 4.2 SE values for nonlinear ALS case study

Algorithm	**Total SE**	**A**	**B**	**Computational Time (s)**
Guess	136.2	80.0	56.2	-
Full DO	135.7	39.5	96.2	14528
Diagonal DO	113.4	58.8	54.6	2488.6
ALS	106.7	59.1	47.6	344.7

Both DO covariance estimation techniques shown in Table 4.2 use the modified DO algorithm to derive the covariance estimates. The full DO algorithm does not assume the covariances are uncorrelated and derives a covariance value for each element of the Q and R matrices. In contrast, the diagonal DO algorithm follows the procedure shown in Chapter 3 and assumes the noises are uncorrelated, thus only fitting the diagonal components of the Q matrix. Overall, the ALS associated state estimates decrease the total SE by 6% when compared to the diagonal DO and 27% for the full DO.

The computational time shown in Table 4.2 is the time required to derive the covariance estimates for the various techniques using the 200 randomized data sets. For this nonlinear system, ALS experiences a sharp decrease in computational time (86%-98%) when compared against DO. This makes sense as DO requires repeatedly running 200 EKF simulations each time the covariance values are updated during a function evaluation. ALS does not require rerunning these simulations and simply needs to minimize the least-squares function at once using all the information obtained from the innovations.

4.3 ALS Application to GOS Process

Following the performance assessment of the ALS algorithm, the method is now applied to the GOS system to build upon and improve the state estimation results shown in Chapter 3. Due to the relatively large size of the *Q* (10x10) and *R* (6x6) matrices, the diagonal ALS technique from Lima & Rawlings (2011) is applied in this work. This diagonal technique acts similar to the DO formulation from Chapter 3 and simplifies the ALS formulation by estimating only the diagonal components of the measurement and process model covariance matrices. Once again, this reduces the least-squares problem from $\sum_{i=1}^{q} i + \sum_{i=1}^{r} i$ variables to $q + r$ variables, thus reducing the computational time. Additionally, for ill conditioned $\mathcal{A}_k$ matrices this technique has proven to be effective (Lima & Rawlings, 2011).

For direct comparison between DO and ALS, both covariance estimation techniques are run in parallel using 200 randomized data sets and the randomization procedure outlined in Figure 3.2. Additionally, both algorithms are initialized employing an initial guess of the measurement and model covariances set to the identity matrix. ALS is run initially using a N_{ALS} value of 35 and derives the Q_{ALS} and R_{ALS} covariance values shown in Tables 4.3 and 4.4. It is important to note that the true covariances are unknown, due to the noise generation procedure from Chapter 3.

Table 4.3 Q_{ALS} value for GOS process

State	**Value**	**Q_{ALS} Position**
Lac	0.0625	(1,1)
Glu	0.1434	(2,2)
Gal	0.0423	(3,3)
Tri	0.0427	(4,4)
Tet	0.0414	(5,5)
Glb	0.0419	(6,6)
Trig	0.0485	(7,7)
Tetg	0.0414	(8,8)
E	0.3225	(9,9)
V	0.0219	(10,10)

Table 4.4 R_{ALS} value for GOS process

Measurement	**Value**	**R_{ALS} Position**
Disaccharide	0.0624	(1,1)
Glu	0.1423	(2,2)
Gal	0.0414	(3,3)
GOS3	0.0414	(4,4)
GOS4	0.0414	(5,5)
V	0.0219	(6,6)

Due to employing the diagonal ALS technique, all matrix positions not listed in Tables 4.3 or 4.4 are set to a value of 0. It is important to note that the covariances presented above are derived using the normalized form of the state variables to balance the large order of magnitude differences in some process states. As a result, when reviewing these covariance values, the amount of noise is relative to the mean state

values. Interestingly, the ALS algorithm seems to apply an equal amount of weight between the model and measurements. This is most easily shown in states *Glu*, *Gal*, and *V* having nearly identical model and measurement covariance weights for their specific states. States not measured directly also have nearly identical Q_{ALS} values as their corresponding isomer (e.g., *Tri* + *Tetg*). By far the largest covariance value is associated with state *E* as this value is approximately an order of magnitude larger than most of the other process states. Intuitively, this makes sense as the ALS algorithm does not have any corresponding measurement for this state variable. As a result, all the noise for this state must come from the model. Additionally, this state influences all other process states so any noise on this state will eventually propagate to the other states.

When applied to the nonlinear GOS system, ALS exhibits significant computational time benefits over DO as shown in Table 4.5.

Table 4.5 Computational time of covariance estimation algorithms

Algorithm	Computational Time
ALS	309 seconds
DO	10157* seconds

* Indicates part of the DO algorithm was parallelized

Using the 200 randomized data sets, ALS can derive covariance estimates in approximately 3% of the computational time of DO. The least-squares ALS objective function is solved using MATLAB's *fmincon* and provides high quality results without having to modify the standard optimizer settings. In contrast, DO is most effectively solved via a hybrid parallelized *particleswarm+fmincon* optimization algorithm.

Additionally, the DO computational time shown in Table 4.5 is heavily favorable towards DO as this is the clock time of the algorithm, which is reduced by using 8 workers during the parallelized *particleswarm* phase of the algorithm. Without parallelization, this DO computational time would increase and further highlight the computational time benefits of ALS which was not parallelized.

DO requires a more complex optimization approach as simply using *fmincon* can often result in the optimizer driving the EKF algorithm to instability (producing NaN state estimate values) during the function evaluations. When this occurs, the optimizer cannot proceed and returns an error to the user. The most effective avenue for remediating this problem is to initialize *fmincon* with the optimization results from an initial preliminary *particleswarm* search (Kim & Lima, 2020). In this DO implementation, a swarm size of 300 with a max iteration count of 5 is used to provide a more reasonable starting point for *fmincon* despite being well below the default settings for MATLAB. For context, the default MATLAB *particleswarm* settings for a 16 variable optimization problem would be a swarm size of 100 with a max iteration count of 3200. As the individual agents in the swarm operate independently when running, parallelization is easily applied to help reduce the computational time of this algorithm.

Overall, it may be possible to implement alternative optimizers or combinations of optimizers to improve DO's computational time. However, this example demonstrates that for nonlinear systems with relatively large models it is intractable to run DO for online applications, such as live covariance updating to reflect changes in the instrumentation calibration or disturbances to the plant. The necessity of having to continuously rerun

EKF simulations for the DO algorithm will be much slower than simply solving the least-squares optimization problem of ALS.

4.3.1 ALS with EKF for GOS Process

To demonstrate the robustness of the ALS algorithm, the covariances shown in Tables 4.3 and 4.4 are used with an EKF implementation on the GOS process. A sample of estimation results in terms of the state variables are shown below in Figure 4.4.

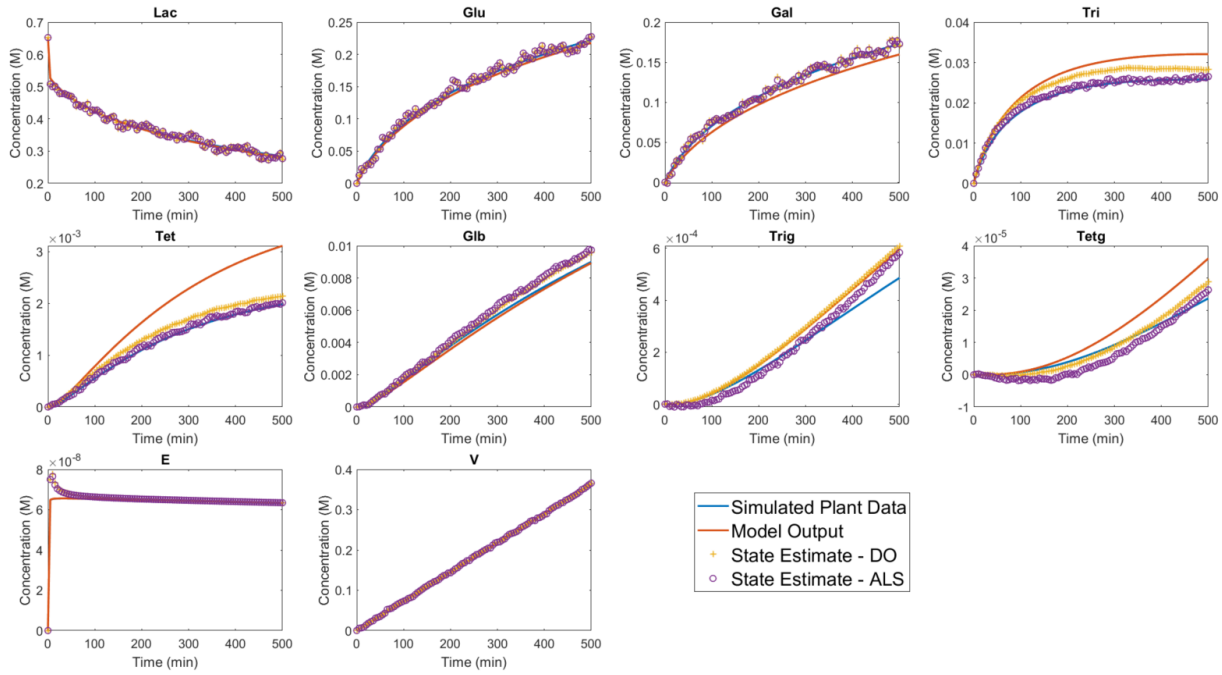

Figure 4.4 EKF performance for state variables using ALS derived covariances

The results shown above in Figure 4.4 follow the conventions from Chapter 3 and plots both the DO and ALS associated EKF results against each other for direct comparison. Overall, the ALS associated results match or exceed the estimation performance as the ones achieved using DO. For states *Lac*, *Glu*, *Gal*, *Glb*, and *V* both sets of covariances produce state estimates that converge to the true process states

(i.e., Simulated Plant Data). For states *Tri*, *Tet*, and *Trig*, the ALS results more closely follow and track the true process states and generate higher estimation performance than the DO associated state estimates. Only state *Tetg* has better estimation performance for DO over the ALS associated state estimates. Figure 4.5 shows the estimation results for the measurements.

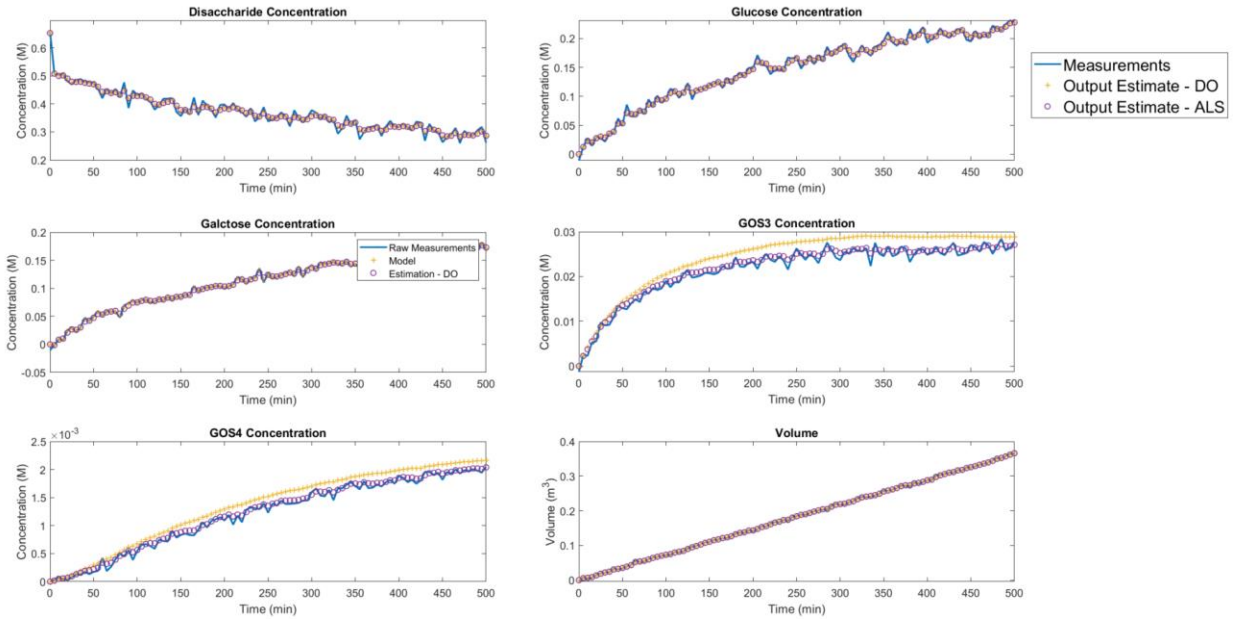

Figure 4.5 EKF performance for outputs using ALS derived covariances

Figure 4.5 also follows the conventions from Chapter 3 and plots both the output estimates from the DO and ALS associated EKFs for more direct comparison. Overall, the ALS associated output estimates closely match or improve the DO associated output estimates. All the ALS associated output estimates lie in-between the peaks and valleys of the white noise corrupted measurements and help to smooth out the noisy data, while for DO the GOS3 and GOS4 output estimates diverge from the trajectory of the measurements. Due to the random stochastic nature of generating the noises, Table 4.6

displays the total SE for both DO and ALS associated state estimates after running 100 EKF simulations using the same datasets for both methods.

Table 4.6 SE values for DO and ALS based EKF estimation

	Lac	*Glu*	*Gal*	*Tri* (10^2)	*Tet* (10^5)	*Glb* (10^3)	*Trig* (10^5)	*Tetg* (10^8)	*E* (10^{14})	*V* (10^2)	Total SE
DO*	1.178	0.259	0.318	2.813	8.447	3.331	2.201	6.820	5.128	1.922	1.806
ALS*	0.856	0.318	0.179	0.314	0.958	4.456	2.037	9.176	6.990	2.862	1.390

*Indicates that one or more simulations were unstable and produced SE values orders of magnitude higher than the average (or NaN simulation results) and thus are not included in the SE totals.

As demonstrated in Chapter 3, EKF based estimation has the potential to fail under different combinations of covariance estimates, data sets, and initial conditions. When generating the results in Table 4.6, both ALS and DO produced 1 infeasible set of estimation results. For most process states, ALS was shown to reduce average SE values, thus demonstrating a higher level of state estimation accuracy. Although several states (*Glu*, *Glb*, *Tetg*, *Et*, *V*) have higher SE values when using ALS, this is insignificant as the total SE is reduced by approximately 23% when using ALS over the modified DO algorithm.

Based upon the qualitative results shown in Figure 4.4 and Figure 4.5 and the quantitative results shown in Table 4.6, the ALS covariance estimation technique can reliably derive valid estimates of the model and measurement covariances that can be successfully used with EKF. These derived covariances ultimately lead to estimation

performance on par with or exceeding DO, while requiring only 3% of the computational to derive the covariance estimates.

4.3.2 ALS with MHE for GOS Process

To further test the robustness of the ALS approach, the derived covariances from both DO and ALS techniques are applied to a series of MHE case studies using the GOS process. A sample of the MHE case study results in terms of the state values is shown in Figure 4.6.

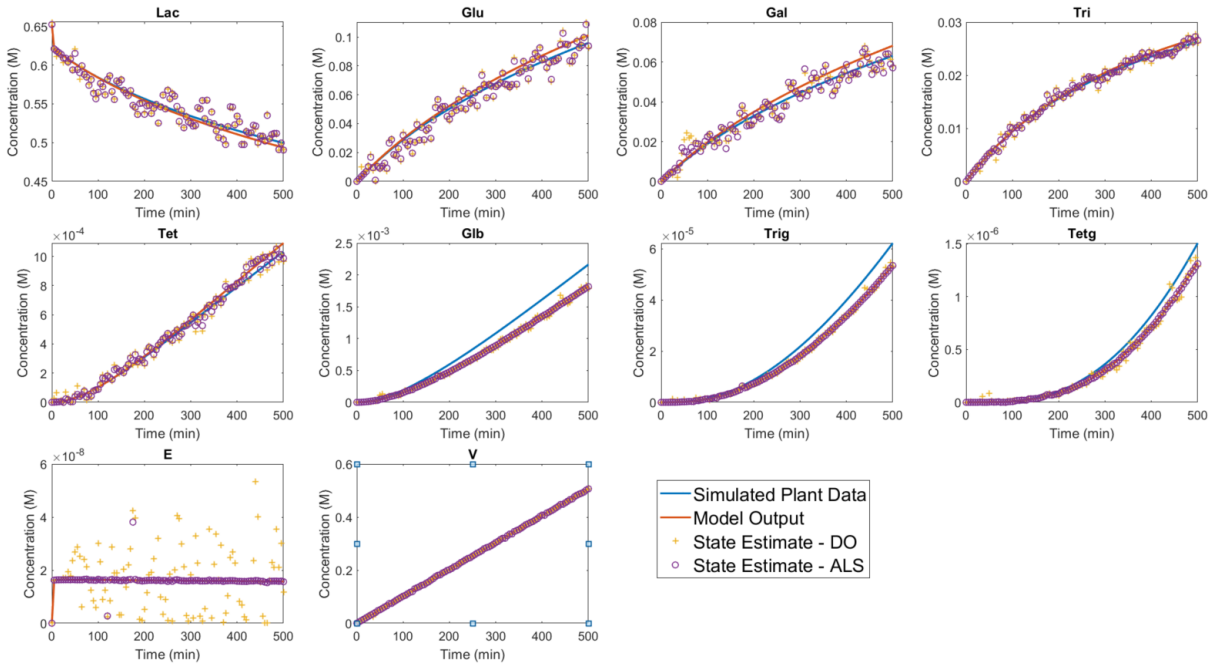

Figure 4.6 MHE performance for state variables using ALS derived covariances

The MHE results shown in Figure 4.6 follow the conventions from Chapter 3 and demonstrate that ALS combined with MHE can generate high quality state estimates. For direct comparison, DO is also applied using the same data sets and consistently fails to generate reasonable state estimates for the enzyme concentration (as demonstrated

with the large rapidly changing scatter dots). This sporadic behavior was also seen in Chapter 3 when using the ad-hoc covariances with MHE. In contrast, ALS closely follows the simulated plant data for the enzyme activity and could be used to accurately infer what the true enzyme concentration is. Beyond this state, the difference between DO and ALS are hard to pinpoint and look similar from a purely qualitative perspective. Other randomized case studies for this process do highlight more robust state estimates using ALS. Overall, the state estimates derived using the ALS covariances closely follow the simulated plant data or the model output. Figure 4.7 shows the corresponding output estimates and the measurements associated with Figure 4.6.

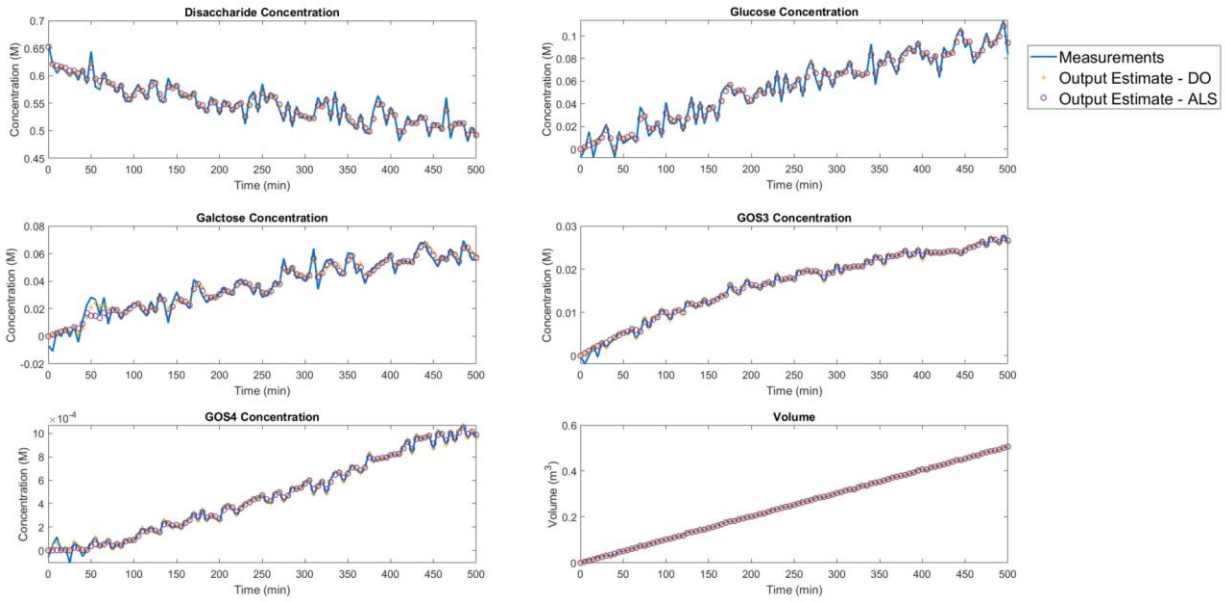

Figure 4.7 MHE performance for outputs using ALS derived covariances

Based upon Figure 4.7, the ALS derived covariances results in an MHE that can filter out the white noise of the measurements as the estimates lie within the peaks and valleys of the measurements. The output estimates for DO and ALS are indistinguishable

in this specific case study but are generally smooth and represent the system well. More noise filtering could possibly be achieved by increasing the length of the MHE horizon, but at the cost of increased computational time. For consistency, these results still utilize the horizon length of 4 from Chapter 3. For a more quantitative analysis of these results, Table 4.7 is provided and is generated using 20 randomized case studies with the MHE using both the DO derived covariances and the ALS derived covariances.

Table 4.7 SE values for DO and ALS based MHE estimation

	Lac	*Glu*	*Gal* (10^2)	*Tri* (10^3)	*Tet* (10^6)	*Glb* (10^4)	*Trig* (10^6)	*Tetg* (10^9)	*E* (10^{13})	*V* (10^2)	Total SE
DO	0.353	0.110	3.794	2.261	4.930	5.500	1.914	8.276	10.71	1.912	0.523
ALS	0.325	0.086	4.331	1.124	2.600	4.827	1.879	8.171	0.498	1.380	0.470

In terms of SE values, ALS performs well and derives system covariances that improves the estimation results for all states, except *Gal*, when compared to using DO. The accurate estimation of enzyme activity shown in Figure 4.5 is consistent across the randomized MHE case studies as the SE for *E* is approximately 95% lower than DO. Additionally, due to the constrained nature of MHE, all the simulations produce valid and feasible estimates across all the randomized case studies. Overall, these results support the conclusion that ALS is an effective covariance approach that is capable of being applied to semi-batch bioprocesses to derive computationally efficient estimates of the process model and measurement noise covariances. As demonstrated, ALS further minimizes the SE of most process states when compared to using the modified DO formulation from Chapter 3 (the most optimal form of DO). For this specific 16 variable

covariance estimation problem, ALS uses 3% or less of the computational time of DO. This significant reduction in computational time could allow ALS to be applied in online settings where recent data sets could be used to update the model and measurement covariances, subject to subtle changes to the plant conditions (e.g., fouling or decreasing activity), measurement offset/bias, or other chronic disturbances. Based upon the MHE results using ALS, application of a more robust algorithm such as Dual-EKF or P-MHE may not be needed for this specific GOS process. In this work, the kinetic parameters are assumed random but held constant across the entire batch time. If instead, the kinetic parameters were allowed to vary with time and subject to a chronic plant model mismatch a more robust algorithm may be required.

Chapter 5 Semi-Batch Process Costing Approach with Dynamic Real-Time Optimization

In this chapter, a semi-batch specific costing approach is developed to strike a tradeoff between individual batch yield, product quality, batch time, and other key factors that influence the profitability of semi-batch processes. A generalized form of this costing approach is discussed and then applied to semi-batch GOS processes using β-galactosidase enzyme from *Kluyveromyces lactis* (*K. lactis*) with economic data obtained from an extensive literature review. As part of this, an enzyme specific cost scaling correlation is developed to approximate bulk enzyme cost from laboratory-scale cost that are more readily available. This semi-batch costing approach is used as the objective function for a novel semi-batch specific DRTO framework. The work presented in this chapter will be published as part of an upcoming manuscript (Alexander, R., Maione, N. R., Ribeiro, M. P. A., & Lima, F. V., In preparation).

5.1 Economic Model and Optimization Approach

Although different reactor configurations may be possible for the synthesis of GOS (e.g., fixed-bed (Albayrak and Yang, 2001), continuous (Shin et al., 1998), immobilized packed-bed reactor (Petzelbauer et al., 2002), batch ultrafiltration membrane bioreactor (Córdova et al., 2016), etc.) this process is commonly performed as a standard batch process at industrial scale (Scott et al., 2016). Conceptually, this method of operation has several challenges such as limited ability to maintain a high substrate (i.e., lactose) concentration, inability to maintain a high enzyme activity, and inability to respond to process disturbances through corrective control actions. Under

batch operation, the substrate (i.e., lactose) concentration and enzyme activity will gradually decline, thus reducing the reaction rate for forming glucose (*Glu*) and galactose (*Gal*) (which are used to form GOS products). These undesired trends can be resolved by switching from a batch to a semi-batch configuration with independent feeding of lactose (u_{lac}) and enzyme (u_{enz}) throughout operation of the reactor. This behavior is supported by analyzing the following equations of the GOS process model (Equations 5.1 and 5.2). The full GOS model considered here can be referenced in Chapter 3.

$$\frac{dlac}{dt} = e\left(\frac{k_H}{K_{MH}} tri - \frac{k_{cat}}{KM} lac - \frac{k_T}{K_{MT}} \gamma\, lac\right) + \frac{u_{lac}C_{lac}}{V} - \frac{u_{in}lac}{V} \tag{5.1}$$

$$\frac{dE_t}{dt} = -k_e E_t + \frac{u_{enz}C_{enz}}{V} - \frac{u_{in}e}{V} \tag{5.2}$$

By gradually feeding fresh lactose and enzyme into the reactor, it may be possible to increase the productivity of the reactor by maintaining a higher production rate of GOS across the batch time. Although not traditionally thought of as a manipulated variable, batch time (t_{batch}) can theoretically be regulated as the control engineer or operator can terminate the batch at any time. This provides an additional process variable (or degree of freedom) that can be varied to improve the process. Overall, this combination of variables and dynamic process nature creates an interesting optimal control problem.

As noted in Chapter 2, it is challenging to optimize the operation of semi-batch processes due to the lack of a steady state, but also the tradeoff between batch yield, batch time, process constraints, and overall process economics. In the literature, Vera et al. (2013) have conducted an optimal control study of a fed-batch (i.e., semi-batch) GOS process using β-galactosidase from *Aspergillus oryzae* (*A. oryzae*) without directly optimizing the underlying process economics. Instead, in their study, the objective

function was the maximization of the reaction yield as a stand in for process economics and was solved using gPROMS (Vera et al., 2013). The work presented in this chapter contributes to the literature in several distinct ways including direct optimization of the process economics, analysis of *K. lactis* derived β-galactosidase for GOS production, Monte Carlo analysis, and derivation of an alternative semi-batch costing methodology. Direct optimization of the process economics will force a compromise between batch time, individual batch performance, and overall GOS production, versus just maximizing the reaction yield.

The main costing function proposed for a generalized semi-batch process is presented as Equation 5.3 and is further broken in the subsequent costing functions presented below that are specific to this GOS process.

$$\max_{t_{batch},u_{j=o,1,\ldots,(t_{batch}/\Delta_{t-1})}} Profitability\ Rate = (Product\ Value - Purification\ Penalty - Feed\ Costs) * (\frac{Batches}{day}) \quad (5.3)$$

The objective function posed in Equation 5.3 seeks to maximize the profitability rate of a general semi-batch process in terms of a $/day rate. This is done via optimizing the batch time (t_{batch}) and feeding profile by manipulating the system inputs (*u*) with a constant discretization time (*Δt*), assuming a zero-order hold is employed for a discrete-time system. Although profitability rate is an unconventional unit for analyzing process economics, for semi-batch processes, this helps to balance the number of batches produced per day and the amount of product generated in each individual batch. The specific costing functions for the GOS process are broken down in Equations 5.4-5.12.

$$Product\ Value = GOS\ Mass * GOS\ Value \tag{5.4}$$

$$GOS\ Mass = [Glb * MW_{2-sacch} + (Tri + Trig) * MW_{3-sacch} + (Tet + Tetg) * MW_{4-sacch}] * V \tag{5.5}$$

$$Purification\ Penalty = Non\ GOS\ Mass * \ Penalty\ Rate \tag{5.6}$$

$$Non\ GOS\ Mass = [(Lac * MW_{2-sacch}) + (Gal + Glu) * MW_{1-sacch}] * V \tag{5.7}$$

$$Feed\ Cost\ = Lactose\ Feed\ Cost + Enzyme\ Feed\ Costs \tag{5.8}$$

$$Lactose\ Feed\ Cost = \left(\left[\sum_{i=0}^{(t_{batch}/\Delta t)-1} u_{lac\ t=i*\Delta t} * C_{lac} * \Delta t\right] + init_{lac}\right) * MW_{2-sacch} * Lactose\ Cost \tag{5.9}$$

$$K.lactis\ Feed\ Cost = \left(\left[\sum_{i=0}^{(t_{batch}/\Delta t)-1} u_{enz\ t=i*\Delta t} * \Delta t\right]\right) * \rho * K.lac.Bulk\ Cost\ * Dilution\ Fac \tag{5.10}$$

$$\frac{Batches}{day} = \frac{1440\ min}{(t_{batch}+t_{prep})} \tag{5.11}$$

The specific costing functions shown above utilize the GOS model from Alexander et al. (2023) and Schultz et al. (2021), but this distinction does not affect the value of GOS products as GOS is simply cost on a mass basis as seen in Equations 5.4 and 5.5. In principle, any GOS model could be used in place of the selected model, but the one chosen has been designed to accurately model enzyme deactivation, *Glb* formation, and GOS formation while preventing overfitting. For dynamic optimization studies, this ability to model enzyme deactivation is critical. The GOS model reports the concentration of

the various components in a molar basis, which is converted to mass using the components molecular weights (*MW*) and the current reactor volume (*V*).

Most GOS processes require downstream purification to meet specific product quality guidelines and can vary significantly depending on the specific market being targeted. For example, GOS can be sold as a powder or syrup with a GOS mass purity range of 48-100 g/100g with varying amounts of galactose and lactose (Lamsal, 2012). The process optimization of these purification steps is beyond the scope of this work but is accounted for by adding a penalty rate for failing to remove non-GOS components, calculated by multiplying the component concentrations (i.e., lactose – *Lac*, galactose – *Gal*, and glucose – *Glu*), by their molecular weights (i.e., $MW_{2\text{-}sacch}$ and $MW_{3\text{-}sacch}$), and the current reactor volume. This penalty represents the additional separation work that would be required for removing these non-GOS compounds.

The analyzed process considers the independent feeding of lactose and enzyme into the semi-batch reactor with the ability to vary flowrates (*u*) throughout the batch. This requires the use of the summations seen in Equations 5.9, and 5.10. For clarity, this work assumes a constant lactose feed concentration (C_{lac}) and constant enzyme feed concentration (C_{enz}) across the entire batch time. To prevent a divide by zero error in the calculations, the reactor is initially charged with a small amount of lactose ($init_{lac}$) with a concentration matching the lactose feed. The dilution factor present in Equation 5.10 scales the cost of the enzyme feed if diluted from the industrial scale concentration, as β-galactosidase from *K. lactis* is typically sold as a concentrated solution. In Equation 5.10, the solution feed density (ρ) is used to convert the volumetric flowrate to a mass flowrate.

Equation 5.11 calculates the number of batches per day and allows the user to assume a preparation time (t_{prep}) to unload, clean, and prepare the reactor for another batch. Without incorporating t_{prep} directly into the optimization problem, there is no penalty for running many short batches per day. The numerator of this equation is the number of minutes of operations per day and can be switched to alternative time units (i.e., hours) if t_{prep} and t_{batch} are dimensionally consistent. Section 5.2 details how these economic parameters were derived for this chapter.

The maximization of the profitability rate through optimization of the batch time and feed rates requires a unique problem formulation as the number of manipulated (i.e., decision) variables change as a function of the batch time and discretization time. Longer batch times increase the number of opportunities to vary the system inputs (i.e., lactose and enzyme flowrates), thus making the total number of manipulated variables dependent on batch time (another decision variable). For this GOS process, Equation 5.12 quantifies the relationship between batch time and the number of manipulated variables.

$$Total\ \#\ MVs = \frac{Batch\ Time}{(\Delta t)} * 2 + 1 \tag{5.12}$$

Conventional optimization algorithms such as MATLAB's *fmincon* (MathWorks, n.d. A), particle swarm optimization (Poli et al., 2007), and IPOPT (Wächter & Biegler, 2006) require a fixed number of variables and are unable to dynamically change the number of decision variables while running. To remediate this issue, this economic study structures the problem using a novel bilayer DRTO approach as shown in Figure 5.1.

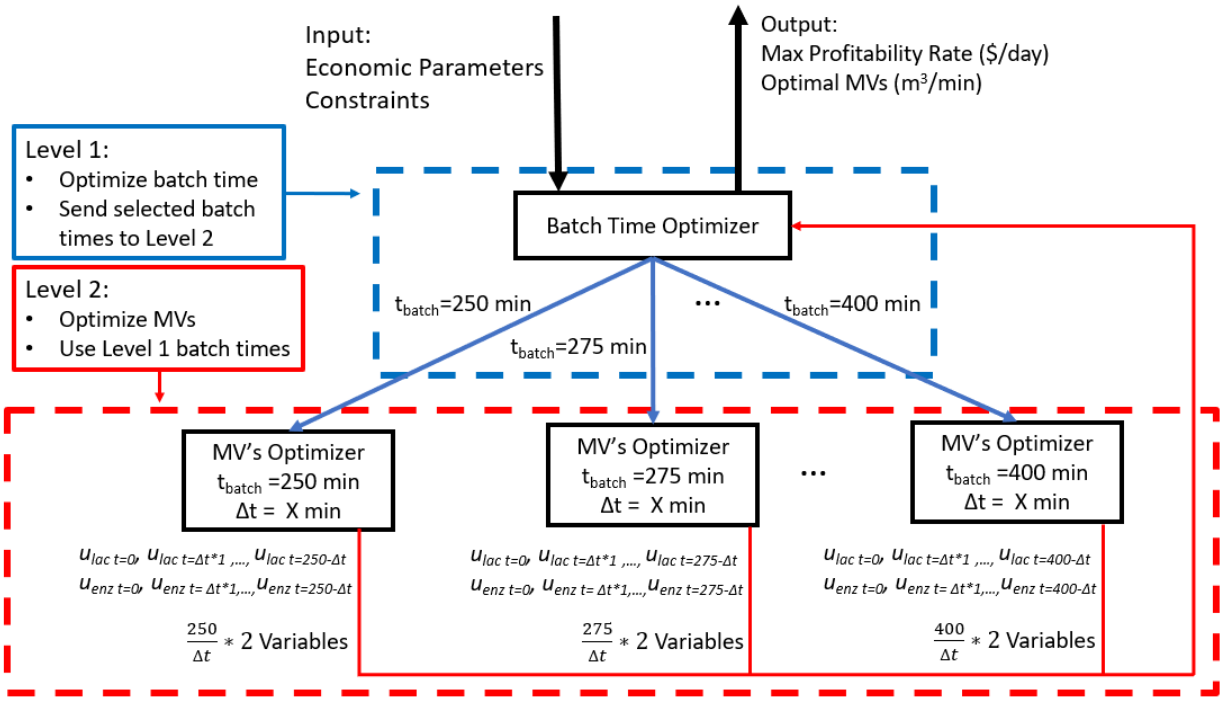

Figure 5.1 Bilayer DRTO approach

This bilayer DRTO approach splits the problem into two distinct layers, Level 1 and Level 2. Level 1 (or the top layer) selects a batch time within a given range of values (i.e., min and max value constraints) and can employ a basic searching algorithm such as golden search or sensitivity studies to efficiently select batch times to be evaluated. Upon selection of several candidate batch times to be evaluated, Level 2 of the DRTO approach is employed. Unlike Level 1, Level 2 hosts several nodes in which each receive a single specific candidate batch time (i.e., one batch time per node). Each node in Level 2 derives the optimal lactose and enzyme flowrates into the reactor at every discretization time for the node's assigned batch time. With each node in Level 2 receiving a fixed batch time, the number of manipulated variables remains constant, thus

allowing standard optimization algorithms to be employed. Once the Level 2 node derives the maximum profitability (and the relevant optimal control inputs) for its assigned batch time, it feeds this information back to Level 1 to inform the search for the optimal batch time, subject to the specific Level 1 optimizer being used. The ultimate output from this bilayer DRTO is a single optimal batch time with the corresponding optimal lactose and enzyme feed flow rates required to maximize the profitability rate.

Although framing the DRTO in this bilayer manner method requires solving for multiple Level 2 nodes, this structure allows parallel computing to be performed where each Level 2 node is simultaneously solved. As MATLAB is being used to develop this bilayer DRTO, parallelization is easily performed using the Parallel Computing Toolbox. Additionally, by employing this bilayer approach, constraints regarding the control moves are isolated to Level 2. These constraints are shown as Equations 5.13, 5.14, and 5.15.

$$0 \leq V(t) \leq V_{max} \tag{5.13}$$

$$V(t) = \sum_{i=0}^{(t_{batch}/\Delta t)-1} u_{lac\ t=i*\Delta t} * \Delta t + \sum_{i=0}^{(t_{batch}/\Delta t)-1} u_{enz\ t=i*\Delta t} * \Delta t \tag{5.14}$$

$$u_{min} \leq u(t) \leq u_{max} \tag{5.15}$$

In which, Equation 5.13 is a linear inequality constraint for the maximum reactor volume (V_{max}) and is related to the control moves using Equation 5.14. The reported literature review did not provide any specific guidance for sizing industrial GOS reactors. However, Scott et al. (2016) did state that batch reactors typically range from cubic meters to 10s of cubic meters. Equation 5.14 simply tracks the amount of lactose and enzyme entering the reactor across the batch time and ensures the reactor is not

overfilled. Equation 5.15 influences these equations by providing minimum and maximum values for the specific control moves. Although not specifically outlined in an equation, the batch times on Level 1 are subject to a minimum and maximum value constraint to assist in bounding the optimization.

In this specific implementation, the bilayer DRTO is being used to derive the optimal feeding policy of a nominal process (i.e., no noise GOS process) and is not rerun subject to the state feedback from the plant. A true DRTO algorithm would continuously update the optimal dynamic trajectory, subject to the current plant conditions. If this bilayer DRTO were to be applied to a real system, the optimal batch time and associated control moves would be updated frequently. As the bilayer DRTO is being applied in this chapter as a tool for evaluating GOS profitability, it is being applied as a dynamic optimization tool exclusively.

5.2 GOS Process Economic Parameters

Table 5.1 summarizes the results of the conducted literature review and provides initial guidance on selecting values for specific GOS economic parameters.

Table 5.1 Economic parameters found in literature for GOS process

Parameter	**Published Values**	**Assumed Value (in this study)**
GOS Value (Future Market Insights, 2020; Yang, 2003; Tang, 2002; Valero, 2009)	$6/kg, $10/kg, $11/kg, $17/kg	$11/kg
Lactose Cost (Tang, 2002; Valero, 2009; Scott et al., 2016; USDA, 2023)	$0.26/kg-$0.88/kg, $0.88/kg, $0.5/kg, $0.66/kg-$1.15/kg	$0.7/kg
β-galactosidase from *K. lactis* Cost (Valero, 2009)	$550/kg	$670/kg-$1000/kg
Penalty Rate	Not found in Literature*	$1/kg
Preparation Time	Not found in Literature	60 minutes

*Scott et al. (2016) conducted an economic analysis of industrial scale GOS processes and reported the associated “Separation aids” contribution to the minimum product selling price (MPSP)

When reviewing the literature on GOS process economics, there were no clear guidelines on how to evaluate these processes or commonly accepted economic parameter values due to the wide range of GOS products and different production pathways. The GOS values shown in Table 5.1 are derived from a variety of sources and did not include any specific mention to the type of GOS product being made. The GOS value of $6/kg is not directly from Future Market Insights (2013) but is derived by dividing their reported GOS market value and their reported metric tons of GOS sold. The lactose cost varies significantly and has been reported as having a volatile market price (Yang,

2003) constantly tracked by the United States Department of Agriculture (USDA, 2023). Due to these fluctuations, it is reasonable to assume a $0.7/kg price for lactose.

The penalty rate is a unique aspect of this costing approach and could not be found directly in the literature. Essentially, this penalty is here to represent the required downstream purification costs of having excess non-GOS components at the end of a batch. After conducting the literature review, the most comparable data found was from Scott et al. (2016). The assumed $1/kg value was developed after reviewing the literature results for a case study producing GOS considering 5 different processes (2 with β-galactosidase from *A. oryzae* and 3 with β-galactosidase from *Bacillus circulans*) with a combination of purification techniques to produce a 75% pure GOS product. Eventually, these case studies derived the minimum product selling price (MPSP) with a breakdown of various categories including “Separation aids”. These values range from approximately $0.04/kg US to $0.54/kg US (Scott et al., 2016). The $1/kg value assigned in Table 5.1 provides an additional safety cushion for determining the profitability rate.

The most challenging values to find in the literature were approximations of the enzyme cost. Generally, most literature did not report their specific enzyme feed costs and simply provided the enzyme used in their experiments. For *K. lactis* derived β-galactosidase, only a single estimate of the enzyme cost could be found in the academic literature, which ended up being a 2009 PhD dissertation from Valero (2009). This reference simply listed the β-galactosidase from *K. lactis* cost as $550/kg with an activity of 3000 lactase unit/g or mL with no specific mention of how this cost was obtained. This study (Valero, 2009) does mention the β-galactosidase from *K. lactis* used in their

experiments was obtained from Valley Research and the specific criteria used to define one lactase unit in their work.

As a result of these challenges, the enzyme cost found in the literature could not be solely relied upon as it was not clear how this $/kg value was derived or how this cost could vary depending on the enzyme provider. For example, it was unclear how a different β-galactosidase from *K. lactis* vendor may price their enzyme given differences in factors such as enzyme feed concentration (i.e., amount of dilution/purification), activity, localized market conditions, or even discount rates for industrial clients. Attempts were made to convert these $/kg values to $/U (or $/activity unit) values to help compare enzymes across different vendors, but enzyme activity is a nonstandard unit often customized to specific enzyme solutions.

Valero (2009) defines one β-galactosidase from *K. lactis* lactase unit (LU) as “the amount of enzyme which liberates 1 μmol of glucose per minute from lactose (concentration 10%) at the early stages of reaction at 40 °C, pH 6”. Schultz et al. (2021) also uses *K. lactis* derived β-galactosidase, but defines one unit of enzyme activity (U), “as the amount of enzymes that produces 1μmol of ONP [o-nitrophenyl] per minute” in a potassium phosphate buffer solution, pH 7, containing specific amount of ONPG [o-nitrophenyl-β-D-galactopyranoside], NaCl and $MgCl_2$. An online vendor (Sigma-Aldrich) lists the β-galactosidase from *K. lactis* activity as being ≥ 2600 units/g with one unit hydrolyzing 1.0 umol of o-nitrophenyl-D-galactoside to o-nitrophenol and D-galactose per minute at pH 4.5 at 30°C (Sigma-Aldrich, n.d. A; Sigma-Aldrich, n.d. B). These different activity definitions provide no clear pathway for simply reapplying the enzyme costs found in the reported academic sources.

As a result, the literature review was expanded to consider chemical supply company websites to find bulk or industrial scale enzyme prices. However, this did not yield any specific bulk enzyme costs for *K. lactis*. Of the companies researched, only Sigma-Aldrich published specific quantities of enzyme available for purchase and their associated cost. The main limitation of simply defaulting to using these costs is the laboratory scale at which these enzymes are being sold as it does not reflect the bulk or industrial scale cost. However, it is possible to extrapolate these laboratory scale costs to bulk prices using cost correlations, data from supply catalogues, and the methodology proposed by Hart & Sommerfeld (1997).

The methodology from Hart & Sommerfeld (1997) was developed to estimate the selling price of commercial chemicals produced in specialty quantities from laboratory supply catalogs. In Hart & Sommerfeld (1997), the industrial cost of twenty-four chemicals including benzoic acid, morpholine, 4-octylphenol, and other specialty chemicals were examined using their specific laboratory pricing and the cost formulas defined in Equations 5.16 and 5.17.

$$P = aQ_{corr}^{b} \tag{5.16}$$

$$\log_{10} P = \log_{10} a + b \log_{10} Q_{corr} \tag{5.17}$$

Equation 5.16 is the P-quantity Q correlation, where P is the unit price of the chemical (units \$/kg), Q_{corr} is the purchase quantity (units g), with a being the intercept and b being the slope of the cost correlation. Equation 5.17 is the logarithmic form of Equation 5.16 and is useful when regressing the correlation parameters. For the chemicals evaluated in Hart & Sommerfeld (1997), the weighted average value of

parameter *b* was found to be -0.7518 when using individual correlations. Additionally, Hart & Sommerfeld (1997) also provided Equation 5.18 to predict the bulk price from a single laboratory price if limited data is available.

$$P_B = P_1(\frac{Q_B}{Q_1})^{b_{avg}} \quad (5.18)$$

Equation 5.18 uses the single unit price (P_1) (units \$/kg) and associated purchase quantity ($Q1$) (units g) along with the average slope value (b_{avg}) and the representative bulk amount (Q_B) to estimate the bulk unit price (P_B).

This cost scaling methodology was adapted in this work to enzymes specifically by surveying Sigma-Aldrich values for various enzymes and deriving an enzyme specific slope value (b_{enz}) that could be used to estimate the industrial scale cost of *K. lactis* and *A. oryzae* derived β-galactosidase. A specific enzyme slope value was desired as the cost of specialty chemicals may scale differently than the enzyme market. Table 5.2 shows the specific enzymes used in deriving b_{enz}, the number of different quantities available for purchase, the price range, and each enzyme's specific slope value and intercept.

Table 5.2 Enzyme data used to derive b_{enz}

Enzyme Name	Entries	Pack Size	Price	Activity	Intercept	Slope
β-Galactosidase from *Escherichia coli* (Sigma-Aldrich, n.d. C)	4	1000 – 15000 units	$110.00 – $999.00	≥500 units/mg protein (500)	7.23	-0.182
Cellulase from *Aspergillus niger* (Sigma-Aldrich, n.d. D)	3	5000 – 100000 units	$59.70 – $420.00	≥0.3 units/mg (0.3)	3.98	-0.349
Pectinase from *Aspergillus niger* (Sigma-Aldrich, n.d. E)	4	5000 – 100000 units	$66.60 – $604.00	≥5 units/mg (5)	4.82	-0.264
Pectinase from *Rhizopus* sp. (Sigma-Aldrich, n.d. F)	3	500 – 5000 units	$68.40 – $501.00	400-800 units/g (600)	4.94	-0.170
Protease from bovine pancreas (Sigma-Aldrich, n.d. G)	4	0.250 – 10 grams	$76.00 – $1210.00	N/A	5.33	-0.250
Protease from *Streptomyces griseus* (Sigma-Aldrich, n.d. H)	3	0.100 – 5 grams	$57.00 – $836.00	N/A	5.44	-0.314
α-Amylase from porcine pancreas (Sigma-Aldrich, n.d. I)	5	500000 – 10000000 units	$55.70 – $408.00	≥5 units/mg (5)	3.40	-0.331

α-Amylase from *Aspergillus oryzae* (Sigma-Aldrich, n.d. J)	3	250000 – 5000000 units	$76.70 – $364.00	≥150 units/mg (150)	4.76	-0.473
α-Amylase from human saliva (Sigma-Aldrich, n.d. K)	3	100 – 2500 units	$65.70 – $570.00	1000-3000 units/ mg (2000)	7.51	-0.375
α-Amylase from *Bacillus* sp. (Sigma-Aldrich, n.d. L)	3	1000000 - 25000000 units	$177.00 – $2440.00	≥400 units/mg (400)	4.92	-0.183
Bromelain from pineapple stem (Sigma-Aldrich, n.d. M)	5	0.01 - 1 kg	$85.40 – $3090	N/A	4.14	-0.223
Ficin from fig tree latex (Sigma-Aldrich, n.d. N)	3	100 – 1000 units	$268.00 – $1530.00	≥1units/mg (1) * (Sigma-Aldrich, n.d. O)	6.18	-0.246
Catalase from *Aspergillus niger* (Sigma-Aldrich, n.d. P)	3	10 – 100 mg	$111.00 – $552.00	N/A	6.44	-0.303
Lipase from *Aspergillus niger* (Sigma-Aldrich, n.d. Q)	3	0.100 – 5 grams	$28.90 – $258.00	N/A	4.99	-0.471
Glucose Oxidase from *Aspergillus niger* (Sigma-Aldrich, n.d. R)	5	10000 – 2500000 units	$62.40 – $3020.00	100000-250000 units/g (175000)	5.56	-0.213

Alcohol Dehydrogenase from *Saccharomyces cerevisiae* (Sigma-Aldrich, n.d. S)	6	7500 – 300000 units	$50.00 – $646.00	≥300 units/mg (300)	5.81	-0.286
Glyoxalase I from *Saccharomyces cerevisiae* (Sigma-Aldrich, n.d. T)	4	200 – 2500 units	$137.00 – $1010	≥400 units/mg (400)	7.75	-0.209

*(Sigma-Aldrich, n.d. O) is required as (Sigma-Aldrich, n.d. N) did not list the activity, so the product specification sheet was consulted for this conversion factor

All 17 enzymes in Table 5.2 were selected as they contained 3 or more entries on their product page which made regression of the slope ($\log_{10} a$) and intercept (*b*) possible. Many of these enzymes were listed in terms of number of units being sold which did not align with the $/kg purchase price and the grams of material purchased required in Equations 5.16-5.18. As a result, the reported activity per mass units were used to properly convert units where needed. The activity per mass units in parenthesis are the values that were assumed to make the conversion. Intercept values in Table 5.2 are reported in their $\log_{10}$ form and vary from 3.4-7.75 depending on the enzyme cost. These values are much larger than the 2.4-4 range reported in Hart & Sommerfeld (1997) as enzymes are typically much more expensive than specialty chemicals.

The slope values found for the enzymes range from -0.17 to -0.473 and have a b_{enz} value of -0.285, which is significantly lower than the -0.7518 b_{avg} for the chemicals examined in Hart & Sommerfeld (1997). This would suggest that enzyme cost is not

subject to the same large bulk price decreases as specialty chemicals are. It may be possible that this -0.285 value is the result of costing formulas specific to Sigma-Aldrich, but without industrial prices for these enzymes it is challenging to confirm this hypothesis. As a result, this derived b_{enz} is assumed to be accurate for implementation in this study. The costing data specifically for β-galactosidase from *K. lactis* is presented below in Table 5.3 and was used to approximate the bulk price of this enzyme (Sigma-Aldrich, n.d. A).

Table 5.3 β-galactosidase from *K. lactis* laboratory costing values (Sigma-Aldrich, n.d. A)

Pack Size	**Price**	**Purchase Quantity**	**Unit Price**	**Bulk Price (From Equation 5.18)**
50 ml	$114.00	50 g	$2280/kg	$971.39/kg
250 ml	$278.00	250 g	$1112/kg	$749.26/kg

Due to the limited amount of data, Equation 5.18 was applied to each of the available pack sizes assuming a bulk purchase quantity of 1000g along with using b_{enz}. Additionally, a solution density of 1g/ml was assumed to convert pack size to the appropriate units for the cost correlation. This produced an estimated bulk enzyme price range for *K. lactis* derived β-galactosidase of approximately $750/kg-$970/kg. When applying b_{enz} and using these two data points, a $\log_{10} a$ value of 3.767 was regressed. This intercept value in conjunction with Equation 5.16 and b_{enz} were used to produce the cost scaling chart shown in Figure 5.2.

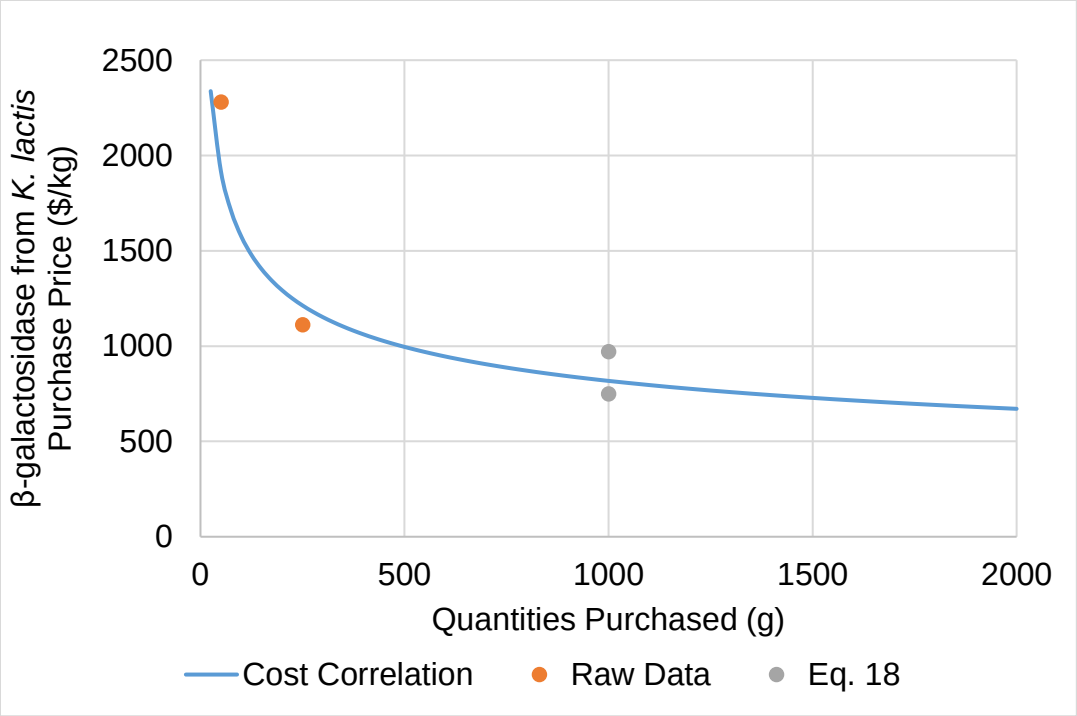

Figure 5.2 β-galactosidase from *K. lactis* purchase price correlation

The raw data shown in Figure 5.2 is obtained from Table 5.3 and closely follows the cost correlation derived using b_{enz}. For additional comparison, the costs scaled using Equation 5.18 are also plotted in Figure 5.2 and agree with the cost correlation. When following the cost trajectory to a purchase quantity of 2000 g, the purchase price begins to level out and approaches a value of $670/kg. As this cost correlation is subject to uncertainty, the values shown in Figure 5.2 are consistent when reviewing the $550/kg β-galactosidase from *K. lactis* cost from Valero (2009). This $550/kg price is lower than the values shown in Figure 5.2 but is from 2009 and has not been adjusted due to inflation. If the standard 41.4% U.S. cumulative rate of inflation from 2009 to 2023 is applied to this value, then the adjusted price would be $777.72/kg (CoinNews Media Group Company, n.d.) which is more in line to the current prices estimated in Figure 5.2.

Overall, between the cost curves and limited historical data it is reasonable to assume a bulk β-galactosidase from *K. lactis* cost of $777/kg.

Although many of these economic parameters are hard to pinpoint down due to regional influences, market volatility, and pricing agreements, the values and methodology presented in this work can serve as basis for an initial economic study to determine profitability of operating GOS processes in a semi-batch mode. Additionally, the framework and economic analysis is designed to be easy to implement and rerun as the underlying process economics shift or additional data becomes available. This identified uncertainty is also a large motivation for conducting the Monte Carlo analysis discussed below.

5.3 Case 1: β-galactosidase from *K. lactis* Results

The application of the economic parameters outlined in Section 5.2 along with the framework outlined in Section 5.1 are employed to generate the results presented. For these performed case studies, a max reactor volume of 1 m^3 was assumed and different inlet lactose concentrations were applied (0.4 M – 0.8 M) to examine their effect on profitability. An economic analysis of different lactose feed concentrations was desired as specific feedstock concentrations can vary depending on the preprocessing steps or originating source. Additionally, all DRTO results were achieved by using MATLAB's *fmincon* optimization algorithm with an increased number of maximum function evaluations of 15000 to improve the quality of the optimization results. Moreover, minimum and maximum constraint values of 1×10^{-8} m^3/min and 1×10^{-1} m^3/min were

applied to the inlet flowrates while implementing the zero-order hold on the system inputs.

5.3.1 Diluted β-galactosidase from K. lactis DRTO Results

When conducting the economic analysis of the diluted β-galactosidase from *K. lactis* enzyme feed, the commercial (i.e., feedstock) grade enzyme solution was diluted from an estimated commercial concentration of 2.7×10^{-4} M to a concentration of 3.63×10^{-7} M (a dilution factor of 0.0013) to be within the range of enzymes concentrations used in the GOS model from Schultz et al. (2021). A diluted enzyme case study is desired as much of the existing GOS literature uses diluted enzyme feed solutions and regress model kinetic parameters using data collected from these conditions. Figure 5.3 shows the specific DRTO results for multiple inlet lactose concentrations across batch times ranging from 100 minutes to 2000 minutes, with a discretization time of 25 minutes and β-galactosidase from *K. lactis* costs of $1000/kg and $670/kg. The $1000/kg value for β-galactosidase from *K. lactis* represents the upper bound of the expected bulk purchase price (i.e., worst case scenario) as seen using Equation 5.18 data from Table 5.3. Also, $670/kg is considered the lower bound of the expected bulk purchase price (i.e., best case scenario) and is taken from Figure 5.2.

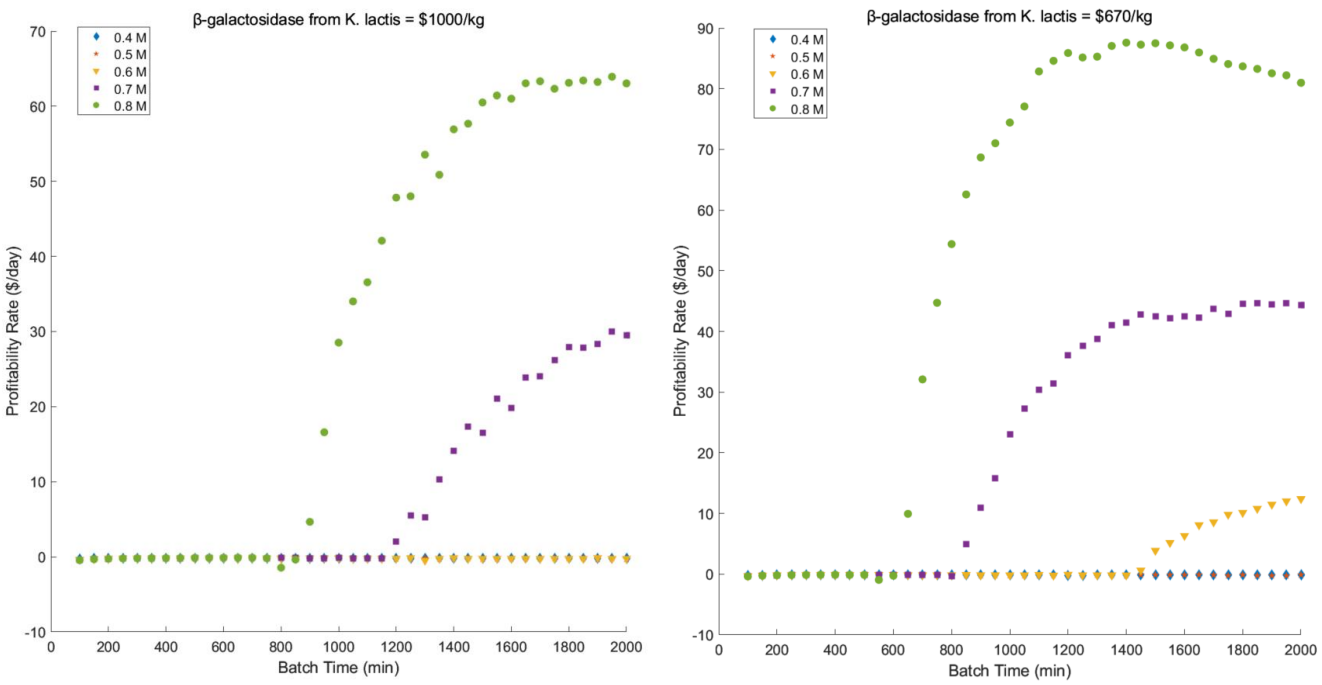

Figure 5.3 Diluted β-galactosidase from *K. lactis* DRTO results

The main takeaway from Figure 5.3 is the importance of lactose feed concentration on profitability potential of the GOS process regardless of enzyme cost. Based upon this economic analysis, lactose concentrations less than 0.6 M are unable to produce positive profitability rates under the two different enzyme prices evaluated. Under the lowest expected bulk cost, 0.6 M lactose feed solutions begin to become profitable given large enough batch times (around 1400 minutes). It should be noted that the DRTO approach used in this work does allow the process to simply not be operated if positive profitability rates cannot be achieved. When combinations of decision variables (e.g., batch time) and parameters (e.g., lactose feed concentration, enzyme cost) are not profitable, the optimizer sets the inlet flowrates to the lower bound (≈0 m^3/min) for the duration of the batch.

A key difference between the tested bulk costs was the maximum profitability achieved and associated optimal batch time. Specifically, the $1000/kg bulk price reached a maximum profitability of approximately $64/day with an optimal batch time of 1950 minutes using a lactose feed concentration of 0.8 M. The $670/kg bulk price achieved a maximum profitability of $87.6/day with an optimal batch time of 1400 minutes using a 0.8 M lactose feed. The associated optimal feeding profiles for each bulk price are shown below in Figure 5.4.

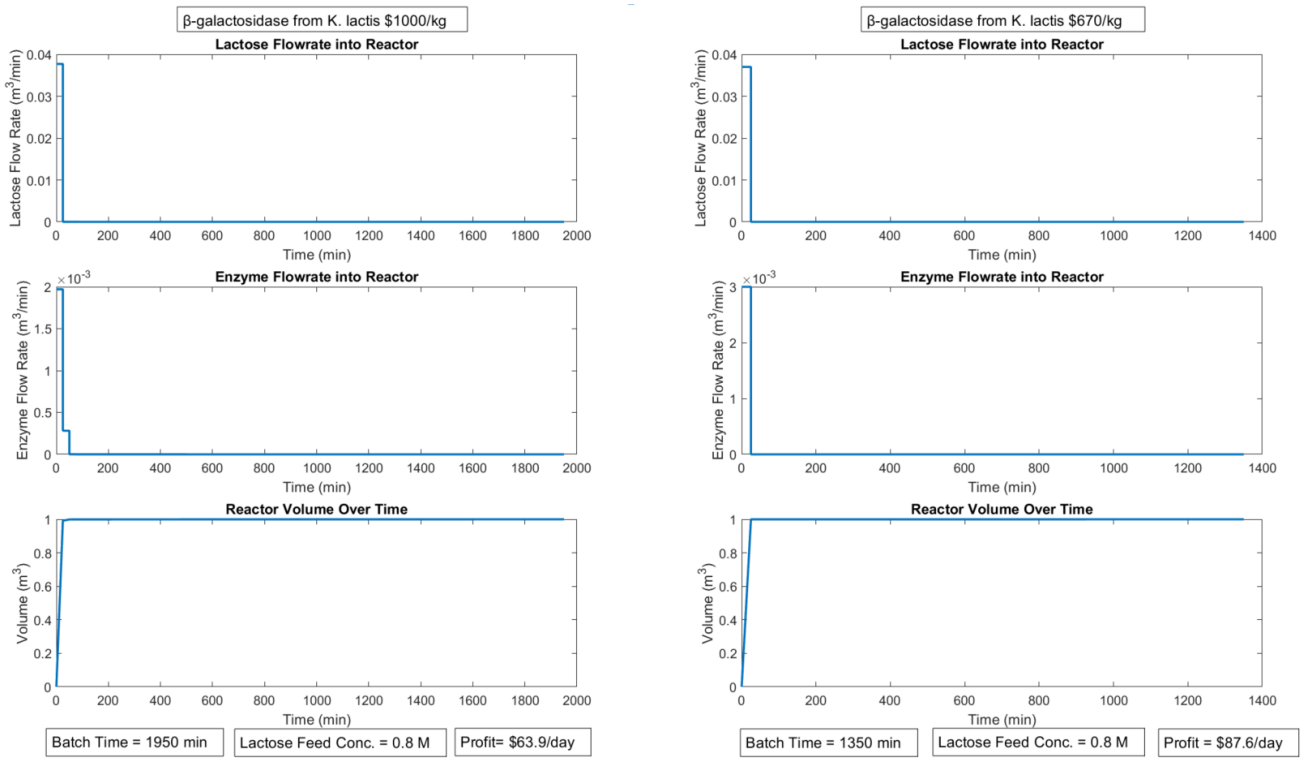

Figure 5.4 Diluted β-galactosidase from *K. lactis* optimal feeding profiles

The results in Figure 5.4 suggest the most optimal way of operating the semi-batch reactor is similar to a batch reactor (with a very brief loading period), regardless of the enzyme feed cost. The left column of Figure 5.4 corresponds to a bulk β-galactosidase from *K. lactis* cost of \$1000/kg and the right column corresponds to a bulk cost of \$670/kg. Within the first control move (0-25 minutes), the DRTO algorithm loads the maximum amount of lactose and enzyme into the reactor and rides the maximum volume constraint (V_{max} = 1 m^3) until termination of the batch. After this set of initial control moves, the algorithm approaches the minimum flowrate constraint for the remainder of the batch. To rule out *fmincon* reaching a suboptimal solution, combinations of initial guesses were assessed. Additionally, varying the enzyme cost seemed to only affect the amount of enzyme added into the reactor during the initial loading period. The \$670/kg enzyme cost used a feed flowrate of 2.9×10^{-3} m^3/min within the first control move, while the \$1000/kg enzyme cost used a feed flowrate of 2.0×10^{-3} m^3/min within the first control move and 2.8×10^{-4} m^3/min during the second control move. The associated component concentration profiles using these optimal feeding profiles are shown in Figures 5.5 and 5.6.

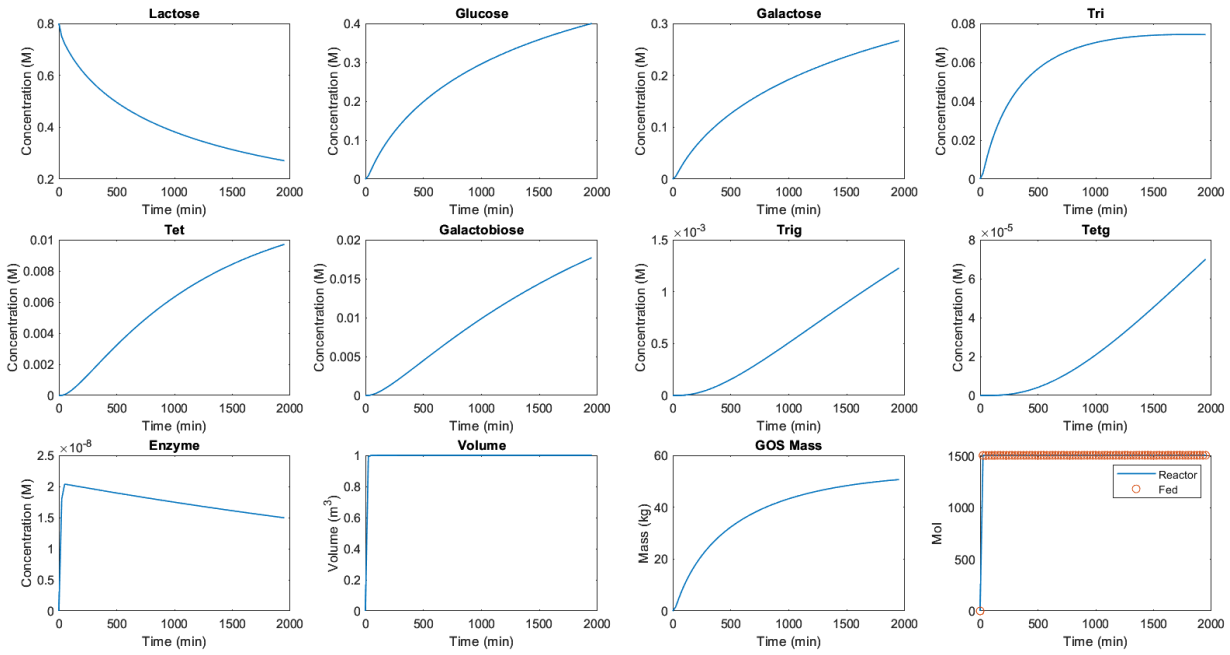

Figure 5.5 Component concentration profiles for optimal diluted β-galactosidase from *K. lactis* profile ($1000/kg)

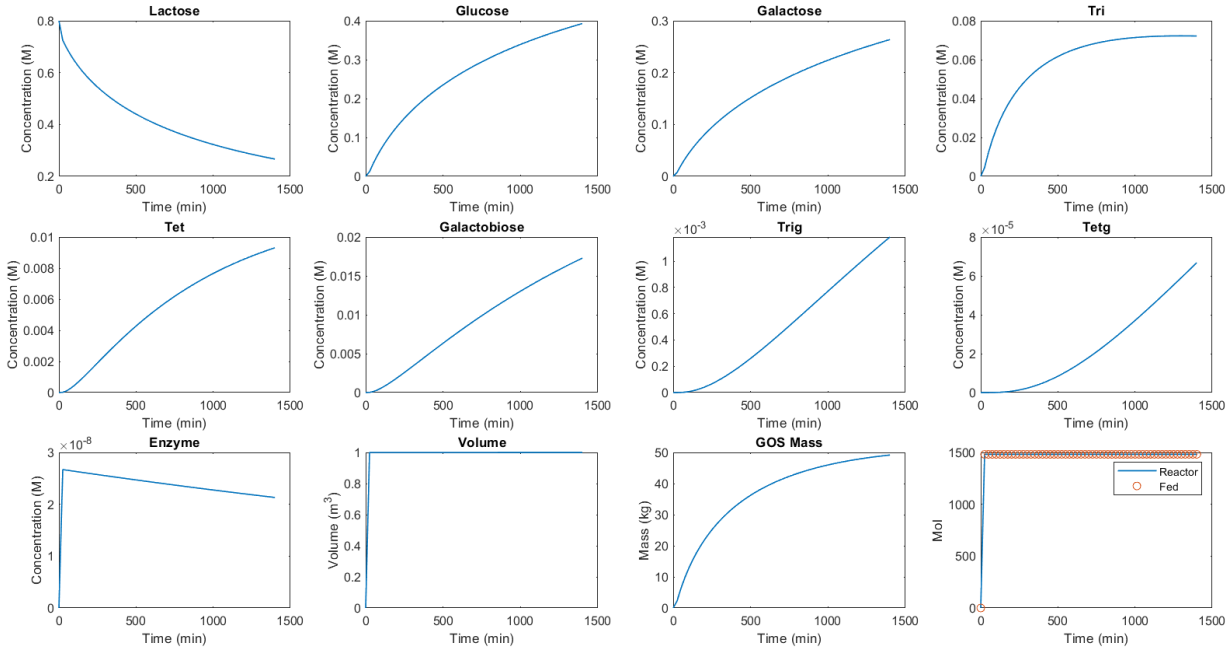

Figure 5.6 Component concentration profiles for optimal diluted β-galactosidase from *K. lactis* profile ($670/kg)

Figures 5.5 and 5.6 track the ten state variables of the GOS model, the amount of GOS mass in the reactor, and the amount of monosaccharide molecules fed into the reactor. The GOS mass is not a state variable in the model but was derived using the process states and Equation 5.5. A monosaccharide molecule balance was performed to help validate the results and tracks the amount of monosaccharide molecules in the system using two different points of calculation. The blue “Reactor” line for the monosaccharide molecule balance is derived by using the component concentrations, reactor volume, and associated number of monosaccharide molecules in each compound (e.g., 2 for *Lac* and 3 for *Tri*). The red “Fed” circles use the inlet lactose flowrates and the inlet lactose concentration to derive the amount of monosaccharide molecules fed into the reactor. Overall, these two data sources match and confirm the mass balance for this system remains solved, thus helping to validate the optimization results achieved.

Despite only having one significant control move into the reactor, the effect of this move can still be seen for the plots in Figures 5.5 and 5.6. For example, in Figure 5.6 the reactor is initialized with 0 M of enzyme and rapidly spikes to a peak of 2.6×10^{-8} M and slowly begins to deactivate as the reaction progresses. In both cases, the GOS mass increases over time with a logarithmic growth style profile and approaches 50 kg of GOS mass as the reaction terminates. Figure 5.5 reaches 50.65 kg and Figure 5.6 reaches 49.17 kg.

As discussed in the previous section, decreasing lactose concentration in conjunction with decreasing enzyme activity was thought to inhibit GOS formation as this would reduce the reaction rate. Generally, the literature has come to the consensus that lactose concentration has a high influence on reaction yield (Vera et al., 2013). The results shown above demonstrate that although the rate of lactose consumption decreases as the batch time increases, the concentration of glucose and galactose continues to rise in the reactor at a stable rate. Theoretically, an elevated enzyme concentration may help to promote the formation of GOS products. However, the prohibitive cost of enzyme forces the DRTO algorithm to strike a balance between GOS formation rate and enzyme activity in the reactor. Overall, this result was unexpected but suggests that there is limited benefit from adopting a semi-batch GOS process over a batch process when considering the underlying economics of a diluted enzyme feed.

5.3.2 Concentrated β-galactosidase from K. lactis DRTO Results

Although the GOS model was not developed using highly concentrated enzyme feeds, additional GOS studies were conducted using the undiluted enzyme feed stream of approximately 2.77×10^{-4} M. This value was derived from the enzyme feed solution used in Schultz et al. (2021) by rationing the diluted enzyme activity (5800 U/L) to the undiluted enzyme activity (6.633×10^{6} U/L) and assuming activity was proportional to concentration. The rationale for using a concentrated enzyme feed solution is that less solvent is required to be fed into the reactor, which reduces the overall dilution of the reactor. Using this new enzyme feed concentration, the DRTO approach was applied to generate the following results shown in Figure 5.7 with a bulk enzyme cost of $670/kg.

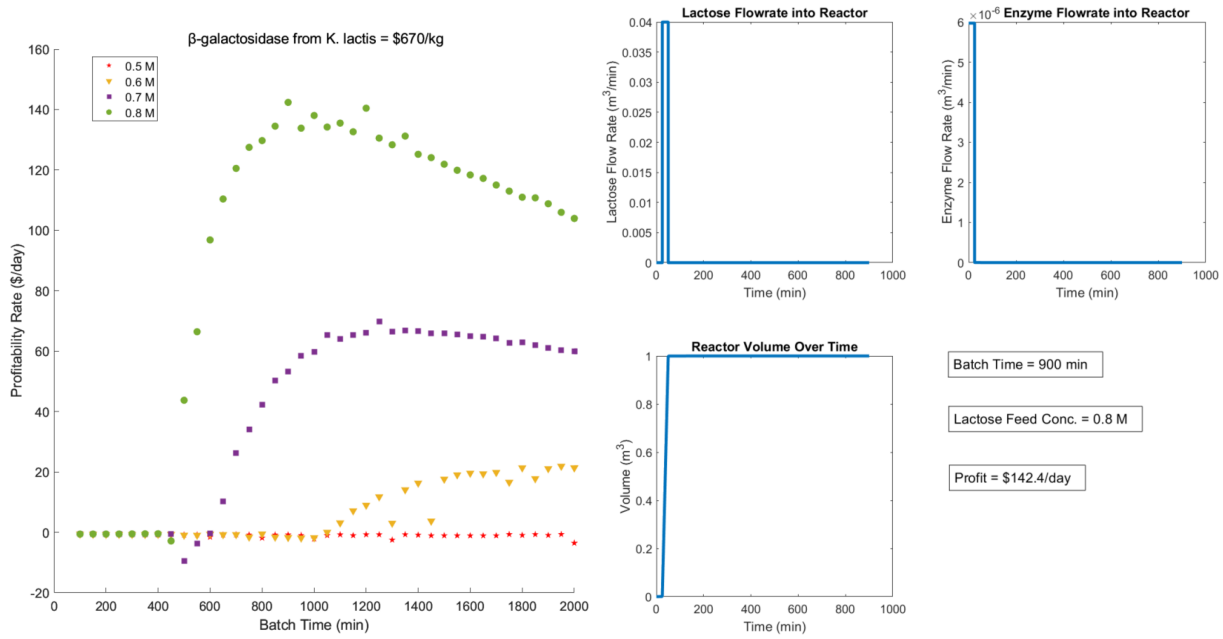

Figure 5.7 Concentrated β-galactosidase from *K. lactis* feed DRTO results

Overall, the results in Figure 5.7 are similar to those from Section 5.3.1 with more concentrated lactose feed streams producing higher profitability rates across the same enzyme concentration. When using the concentrated enzyme feed, the maximum profitability rate achieved was $142.4/day using a batch time of 900 minutes and the 0.8 M lactose feed. The optimal feeding policy results match the diluted case with all the lactose and enzyme being fed within the first or second control moves. Specifically, during the second control move (25-50 minutes), lactose is fed into the reactor at a flowrate of approximately 0.04 m^3/min while enzyme is fed during the first control moves (0-25 minutes) with a flowrate of 6.0×10^{-6} m^3/min. The enzyme flowrate into the reactor is much lower than the diluted case as the solution is approximately 762 times more concentrated than in Section 5.3.1. Additionally, as the industrial enzyme solution is undiluted in this study the dilution factor becomes 1. This raises the cost of the enzyme

feed on an equivalent $/m^3 basis and severely restricts the amount of enzyme solution fed into the reactor.

In this case, some of the 0.6 M results in Figure 5.7 around a batch time of 1600 minutes converged to the suboptimal solution of simply not operating the batch, despite nearby batch times producing positive profitability rates. The most probable cause of this behavior was the initialization conditions of *fmincon* and how minor changes in the enzyme flowrate, enzyme concentration, and GOS production affected the process economics. The results shown here were generated using MATLAB's *GlobalSearch* function with *fmincon* still being selected as the optimization method. *GlobalSearch* uses a scatter-search mechanism to select alternative sets of initial conditions and attempts to find the global minimum by resolving the optimization problem with alternative starting points (MathWorks, n.d. B).

Overall, this posed optimization problem is difficult to solve as the lactose feed and enzyme feed flowrates operate at different orders of magnitude. Additionally, minor changes in enzyme feed flowrate will result in substantial changes to the enzyme concentration, thus affecting the synthesis of GOS. These factors could explain why the optimizer converged to a solution where lactose was fed during the second control move versus the first control move. The sensitivity of this optimization problem is reflected with the component concentrations shown in Figure 5.8.

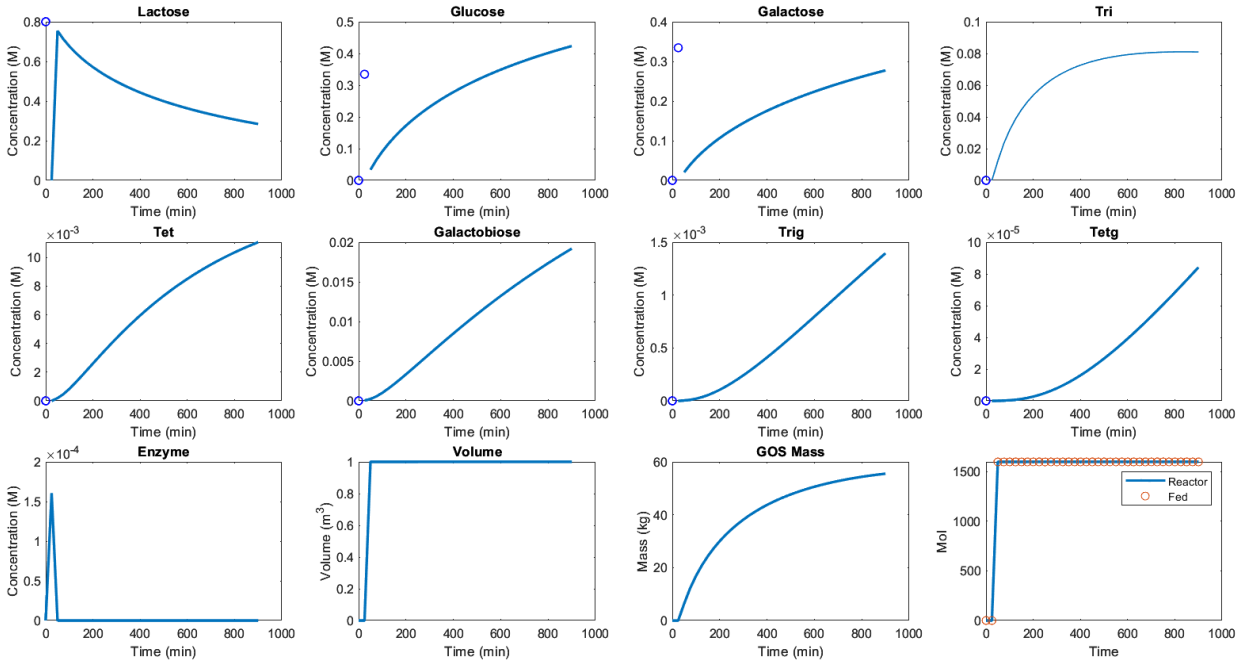

Figure 5.8 Component concentration profiles for optimal concentrated β-galactosidase from *K. lactis* feed

The component profiles shown above in Figure 5.8 generally follow the results shown in Figure 5.6 for most curves once the loading/charging phase is completed. Due to the sensitivity of the optimization formulation, the blue dots in Figure 5.8 are points that were neglected from the component concentration profiles (*Lac*, *Glu*, *Gal*) as they were subject to very rapid changes that are simply present because of potential numerical issues regarding initialization of the batch as it is very stiff and requires using stiff ODE solvers. For example, the reactor is initialized with a small amount of lactose (to prevent a divide by zero for the reactor volume) but very rapidly decreases to a concentration of zero upon enzyme being added into the reactor during the first control move. During the second control move, the much larger lactose feed flowrate rapidly

restores the lactose concentration. This same reasoning explains the very rapid changes in reactor enzyme concentration.

Despite this brief period of rapid changes, the overall component profiles are smooth and produce more GOS (55.5 kg) while reducing the optimal batch time by 500-minutes. This is simply due to the reactor enzyme concentration being much higher than the diluted case as shown in Figure 5.9.

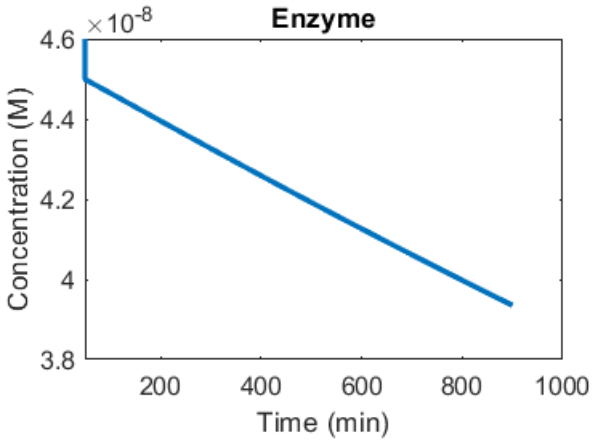

Figure 5.9 Reactor enzyme concentration for optimal concentrated β-galactosidase from *K. lactis* feed

Figure 5.9 is derived from Figure 5.8 but excludes the rapid changes in reactor enzyme concentration to highlight the enzyme concentration present during the synthesis of GOS. The reactor enzyme concentration stays within the 4×10^{-8} – 4.5×10^{-8} M region, which approximately doubles the reactor enzyme concentration seen in the diluted case. This increase in enzyme concentration raises the reaction rate of all reactions and reduces the amount of time it takes to produce the same amount of GOS. In turn, this helps to increase the profitability rate of the reactor.

Overall, this case study suggests that regardless of enzyme feed concentration, the most optimal method of operation is to simply charge the reactor quickly and then operate as a batch process. Additionally, this case study demonstrates that simply feeding the industrial enzyme solution into the reactor is the most optimal way and no dilution is required. This conclusion assumes though that the GOS model and kinetic parameter values accurately represent the synthesis of GOS while operating at these elevated enzyme concentrations.

5.3.3 β-galactosidase from K. lactis Monte Carlo Results

As a result of the uncertainty and wide range of published economic parameters in the literature, a Monte Carlo study was conducted to provide an outlook on how changes in the assumed values could modify the profitability rate and optimal batch time. Due to the large computational time of running hundreds of different case studies, each consisting of tens of Level 2 optimization problems, a modified DRTO algorithm for β-galactosidase from *K. lactis* was developed, validated, and implemented. The modified DRTO algorithm simplifies the problem structure by only optimizing the charging of the reactor (with a zero-order hold of 5 minutes) and then operating as a batch process until terminating the reaction as suggested by the previous results. Simply stated, the simplified DRTO algorithm only optimizes three variables: lactose flowrate from 0-5 minutes, enzyme flowrate from 0-5 minutes, and batch time. Once this loading period is complete, the reactor continues to be operated until the batch time is reached. A mathematical representation of this simplified DRTO approach is shown below in Equations 5.19-5.21.

$$\max_{t_{batch}, u_{lac\ 0\to 5\ min}, u_{enz\ 0\to 5\ min}} Profitability\ Rate = (Product\ Value - Purification\ Penalty - Feed\ Costs) * (\frac{Batches}{day}) \quad (5.19)$$

$$(u_{lac\ 0-5\ min} + u_{enz\ 0-5\ min}) \times 5 = V_{max} \quad (5.20)$$

$$u_{min} \leq u(t) \leq u_{max} \quad (5.21)$$

This simplified approach was developed after reviewing the results from the previous sections and concluding that this process is best operated as a batch when attempting to maximize process economics. As a result, the only optimization required is the amount of lactose and enzyme used to charge the reactor and the total batch time (t_{batch}). Additionally, the previous results suggested that the process is optimized by filling the reactor to V_{max} as all volume curves hit or approach 1 m^3 after the initial control moves. To further simplify the optimization problem, the inequality volume constraint from Equation 5.13 was converted to an equality constraint as shown in Equation 5.20. To validate these modifications to the algorithm, the modified DRTO was run using the dilute β-galactosidase from *K. lactis* conditions with a bulk enzyme cost of $670/kg to produce the optimization results shown below in Figure 5.10.

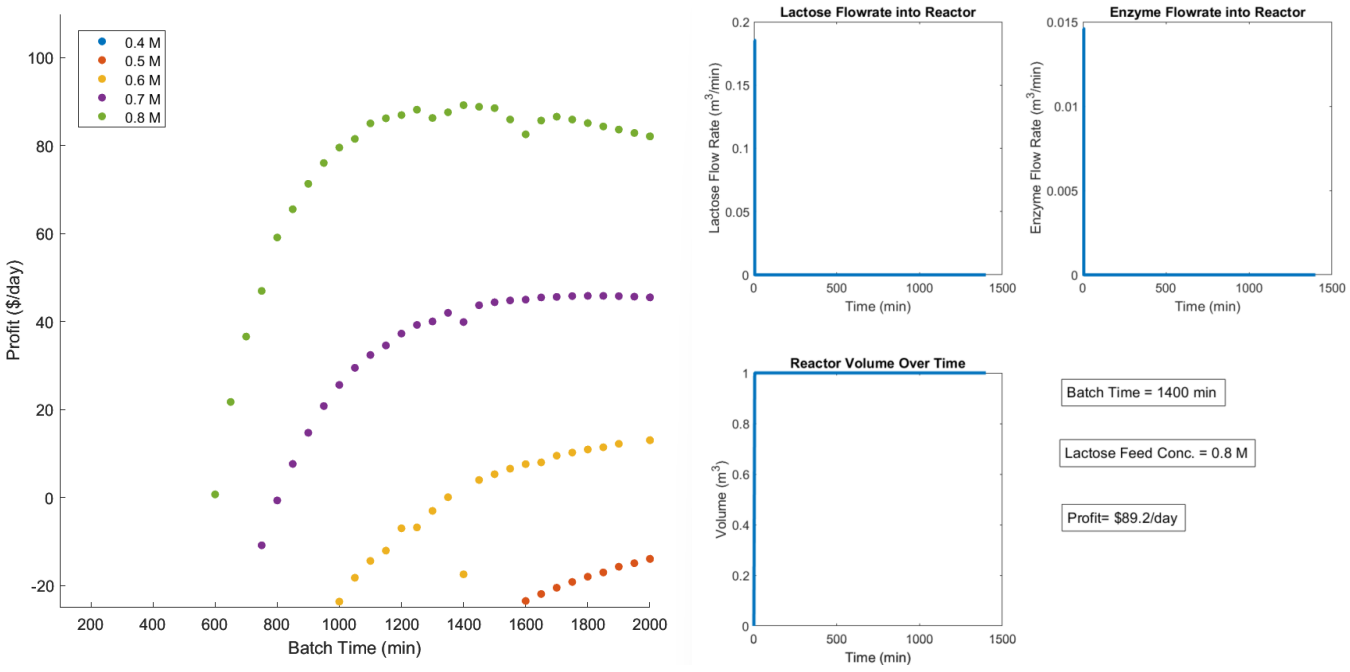

Figure 5.10 Validation plots for modified DRTO algorithm

The results shown above in Figure 5.10 match the results from Figures 5.3 and 5.4 as the maximum profitable rate was achieved using a lactose feed solution of 0.8 M, a batch time of 1400 minutes, and the same quantity of enzyme being fed into the reactor (0.072-0.073 m^3). Although there was a marginal increase in maximum profitability to \$89.2/day when compared to \$87.6/day, this increase is simply due to the reactor being charged with a higher flowrate, thus allowing V_{max} to be reached earlier and more material reacting sooner. The main difference between the results from this modified DRTO approach and the standard DRTO approach was the removal of the algorithm's ability to simply not operate the reactor if a positive profitability rate could not be achieved. Due to the equality constraint, the reactor must be filled to V_{max}, thus producing highly unprofitable simulations at low batch times and low lactose feed concentrations. For easy validation, the y axis of Figure 5.10 is clipped to a lower value of -\$20/day. For

the purposes of the Monte Carlo analysis, only the most optimal simulation results are recorded so this is not an issue.

The Monte Carlo analysis varied the economic parameters in Table 5.1 with an assumed β-galactosidase from *K. lactis* bulk price of $777/kg, where all parameters were subjected to a normal distribution with a standard deviation equal to 25% the nominal parameter value. All economic parameters are varied concurrently and randomized for each of the 400 Monte Carlo iterations. Once again, the effect of lactose concentration was examined by evaluating four inlet lactose concentrations (0.35 M, 0.5 M, 0.65 M, and 0.8 M) with the diluted enzyme feed. Figure 5.11 shows the distribution of the economic parameters and Figure 5.12 shows the resulting distributions of profitability rate and optimal batch time. For these Monte Carlo simulation results, only the most optimal profitability rate and associated optimal batch time were recorded for each inlet lactose concentration at every iteration.

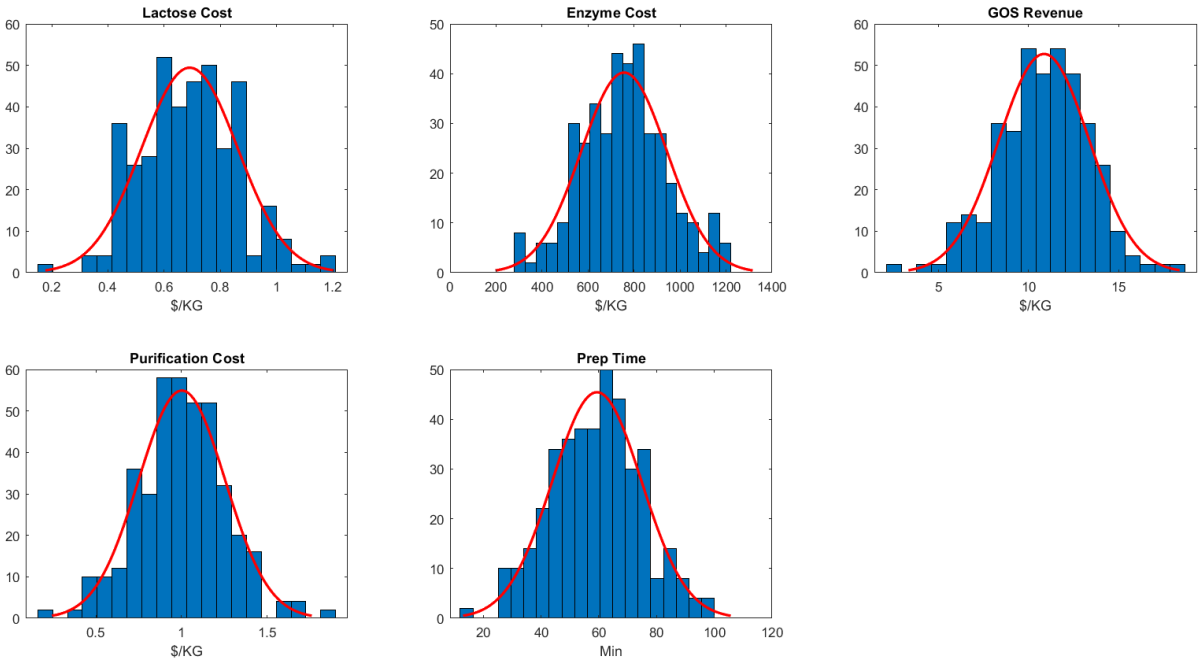

Figure 5.11 Distribution of economic parameters for β-galactosidase from *K. lactis* Monte Carlo analysis

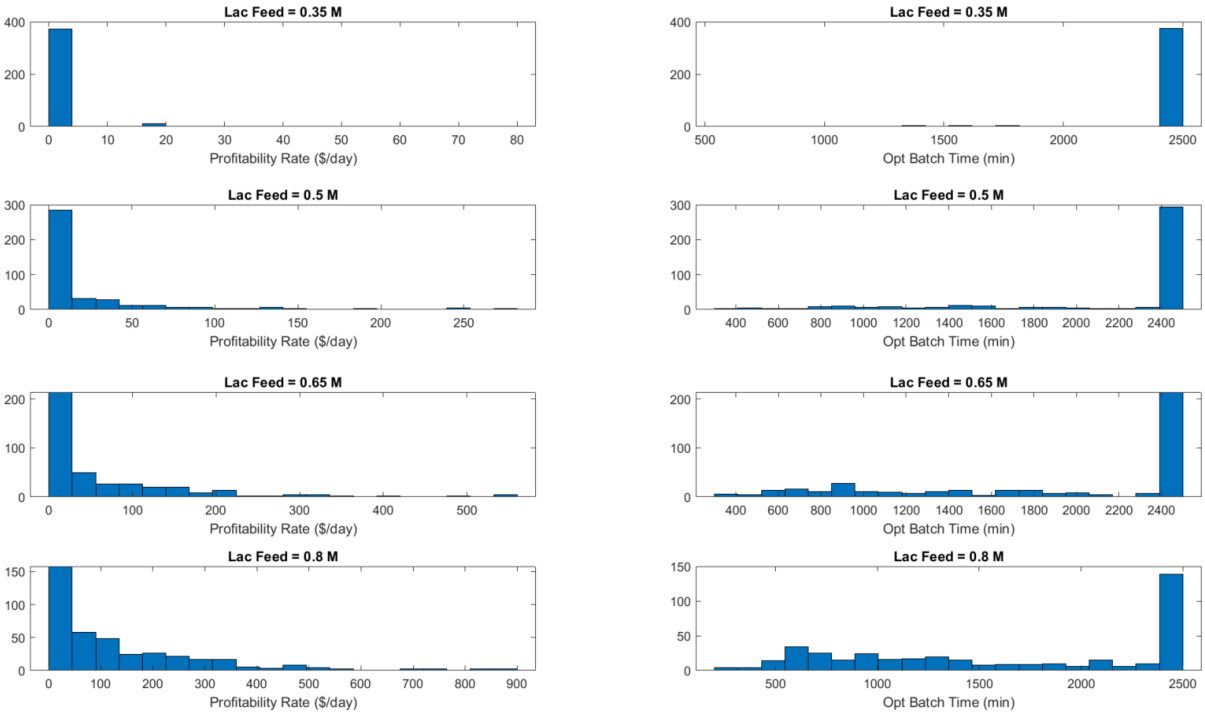

Figure 5.12 Distribution of β-galactosidase from *K. lactis* profitability rate and optimal batch time

The main takeaway from Figure 5.12 is the difficulty in achieving consistent positive profitability rates across all lactose feed concentrations evaluated. This is demonstrated with the large histogram bins located at or below $0/day profitability rate and at a 2500-minute optimal batch times. Profitability rates less than $0/day, were lumped into the $0/day histogram bins of Figure 5.12 to easily show how many simulations were unable to achieve a positive profitability rate. Of the 400 Monte Carlo simulations, 52, 153, 266, and 322 simulations have positive profitability rates for the 0.35 M, 0.5 M, 0.65 M, and 0.8 M inlet lactose concentrations, respectively. As the Monte Carlo analysis randomly varied the economic parameters, it created combinations of parameter values that were unprofitable to operate under. For these simulations, the DRTO algorithm simply decided to operate the batch until the 2500-minute maximum batch time was reached. Due to batch time being in the denominator of the profitability rate expression, when the profitability rate becomes negative, longer batch times allow the rate to approach a value of $0/day. Using the results from Figure 5.12, the average profitability rates were $2.25/day, $19.28/day, $62.72/day, $131.72/day for the 0.35 M, 0.5 M, 0.65 M, 0.8 M lactose feed concentrations, respectively. For these average profitability values, any simulation with a profitability less than 0 was set equal to zero.

Once again, the less concentrated lactose feeds produced worse profitability results as demonstrated by the distribution of profitability rates. The 0.35 M lactose feed had a much higher number of simulations failing to produce a positive profitability and a lower maximum profitability rate than the 0.65 M and 0.8 M lactose feeds. A more detailed quantitative analysis of these results is presented in Table 5.4.

Table 5.4 Quantitative analysis of β-galactosidase from *K. lactis* Monte Carlo results

	Lactose Feed ($/kg)	**Enzyme Feed ($/kg)**	**GOS Revenue ($/kg)**	**Purification Cost ($/kg)**	**Preparation Time (min)**
Monte Carlo	0.690	756.6	10.86	1.002	59.3
Lac. = 0.35 M	0.560	670.9	13.48	0.748	56.5
Lac. = 0.5 M	0.625	698.6	12.87	0.898	60.9
Lac. = 0.65 M	0.661	738.2	12.02	0.952	61.1
Lac. = 0.8 M	0.676	750.2	11.59	0.961	60.0

The objective of Table 5.4 is to help infer the impact each economic parameter has on GOS profitability. This is accomplished through finding the average parameter value across Monte Carlo simulations where positive profitability rates are achieved. Large deviations between an average economic parameter value across all Monte Carlo simulations and the positive profitability rate associated value indicate that this economic parameter plays a large role in affecting the profitability. Theoretically, if the positive profitability rate associated average parameter value matches the Monte Carlo average, then it demonstrates that modification of this parameter will not significantly disturb the ability to achieve a positive profitability rate.

The Monte Carlo row of Table 5.4 contains the average parameter value across 400 simulations. The remaining rows of Table 5.4 track the average parameter values, where positive profitability rates are achieved. As each lactose feed concentration (0.35 M, 0.5 M, 0.65 M, and 0.8 M) is examined independently in the Monte Carlo analysis, each concentration has its own set of average values.

Based upon the simulation results, the preparation time had the least effect on determining when the process became profitable as there is negligible % difference

(-4% - +2%) between the average Monte Carlo value and those for the other lactose concentrations. The most significant economic parameter was the GOS revenue as it has a % difference of +24%, +18%, +11%, and +7% for the 0.35M, 0.5 M, 0.65 M, and 0.8 M inlet lactose concentrations, respectively. The second most impactful economic parameter was purification cost with a % difference of -25%, -10%, -5%, and -4% for the same order of inlet concentrations. These substantial changes in average parameter values suggest these are the most impactful economic parameters that should be optimized when possible.

Chapter 6 Conclusions

The primary aims of this dissertation were to develop state estimation, covariance estimation, and DRTO tools for semi-batch specific bioprocess applications subject to plant-model mismatches. Each one of these individual aims was addressed in a specific chapter of this dissertation and applied to a GOS process to test the viability and performance of these PSE tools. In terms of state estimation, a variety of recursive and optimization-based algorithms were developed and applied successfully. For recursive estimation techniques, EKF and Dual EKF algorithms were implemented and tuned using a preliminary covariance estimation technique known as DO. A specific aspect of novelty in this work was the development of the P-MHE algorithm which modifies the traditional MHE formulation by estimating the kinetic parameters and process states concurrently while reducing the computational time of standard MHE algorithms. When applied to the GOS process, it demonstrated excellent state estimation results and was able to successfully estimate the unmeasured process states from the online measurements and process model.

Upon completion and implementation of these estimation algorithms, covariance estimation techniques such as DO and ALS were investigated to improve estimation performance. In this dissertation, modifications to the published DO formulations were introduced to strike a more fine-tuned balance between the model prediction and plant measurements over the existing formulations. Although this modified DO algorithm derived process model and measurement noise covariances that resulted in lower SE values than the traditional DO algorithms, the associated computational time cost was

very large (i.e., hours). As a result, it would be infeasible to apply this modified DO technique for online covariance estimation applications and thus limits the potential for industrial applications.

To remediate this large computational time, a MATLAB specific ALS code was developed, validated, and applied to derive covariance estimates for the GOS process. The previously available ALS codes were not directly compatible with MATLAB, so modifications to the code were required to produce a MATLAB specific version. When implemented to the GOS system, ALS reduced the covariance estimation time by 97% (approximately 164 minutes), while retaining excellent state estimation performance when coupled with recursive and optimization-based state estimation algorithms. As a result, the developed ALS code could allow for the online covariance updates and improvements to state estimation applied to process monitoring applications.

Finally, the last major aim of this dissertation was to develop and apply a semi-batch specific costing approach combined with a novel semi-batch DRTO algorithm. The profitability rate-based costing function derived for semi-batch processes blends key metrics such as individual batch yield, product quality, and batch time into a single customizable expression. This costing function was used as the objective function of the posed semi-batch DRTO algorithm which employs a bilayer approach to decouple the optimal batch time and optimal control moves into separate layers. This allows for traditional optimization algorithms to be employed to optimize the operation of a semi-batch process. Although the bilayer structure could increase the computational time, the structure easily allows for parallelization to be employed to minimize such time. These developed approaches were applied to the GOS process and ultimately produced the

result that maximum profit is achieved through batch operation versus semi-batch operation when using *K. lactis.*

A byproduct of the DRTO application on the GOS process was the development of a specific cost scaling correlation for enzymes. This correlation was derived from an extensive literature review of laboratory scale pricing and resulted in an enzyme cost scaling slope value of -0.285. This value can be applied to other bioprocesses to assist in conducting economic studies where no bulk purchase price information is available. For the GOS process this correlation led to a bulk purchase price estimate of $670/kg-$1000/kg for *K. lactis* derived β-galactosidase.

Chapter 7 Future Work

In this chapter, several possible avenues for continuing this research are discussed, including some preliminary results for some of these avenues.

7.1 Combining State Estimation, Covariance Estimation, and DRTO

As mentioned in Chapter 2, there are few publications in the literature that combine these various PSE tools together into a cohesive framework. Ideally, the synergistic combination of these tools would follow the steps below to leverage the benefits of each tool. A schematic of the how these tools could interact is shown in Figure 7.1.

1. Covariance estimation techniques such as ALS would use recent sets of plant data, live plant data, or selected sets of historical data to derive accurate covariance estimates of the current plant conditions.
2. State estimation algorithms would utilize the current covariance estimates to derive online and accurate estimates of the process states, including difficult to track states (e.g., related to fouling, coking, efficiency).
3. DRTO algorithms would use the live state estimates to modify the optimal control moves and maximize profitability subjected to the tracked disturbances.

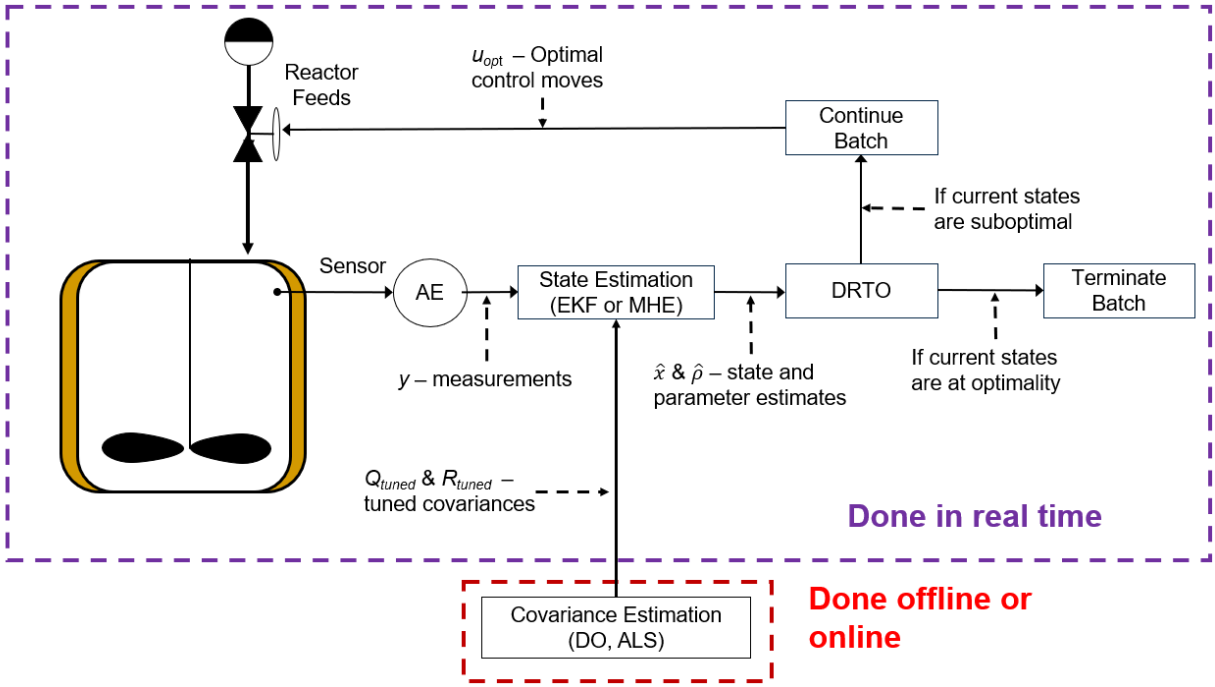

Figure 7.1 Synergistic combination of PSE tools

The DRTO results from Chapter 5 demonstrated that the most optimal operation method for producing GOS is as a batch process, so there was no motivation to build upon and apply this avenue of research to this specific process. Ideally, Figure 7.1 would be best applied to another semi-batch process undergoing process disturbances where the DRTO algorithm could update the optimal control moves subject to the derived state estimates. Although not a bioprocess, the production of polyvinyl chloride (PVC) is traditionally a difficult to control highly nonlinear process that is often conducted using a suspension method (similar to a semi-batch process) (Gau et al., 2018). Additionally, it may be possible to apply this framework to power plant cycling problems as many baseload fossil fuel power plants operate dynamically and undergo frequent transitions between steady states. The dynamic optimization of baseload power plants was

addressed extensively by Dr. Rebecca Kim (2021) and could be coupled with state estimation algorithms to improve the accuracy of the generated DRTO trajectories.

For industrial implementation of this proposed framework, it would be possible to run this algorithm in a dedicated MATLAB or Python application. Communication protocols such as OPC could be used to pull live, recent, or historical data from the DCS or data historian and import the data into the application. By isolating this framework to an external application, the DCS would still be able to safely regulate the process if an error arose with any block of this framework (e.g., memory overload). This imported data could be used by the ALS algorithm to derive updated estimates of the system noise covariances and by the state estimation algorithm to derive live state estimates. Upon generating the live state estimates, the DRTO block could be run to update the optimal control moves. Upon deriving a new optimal trajectory, the MVs could be sent back to the DCS through the same communication protocol and implemented in the plant.

7.2 Applying ALS to Joint State and Parameter Estimation Algorithms

Currently, ALS has exclusively been combined with traditional state estimation approaches (e.g., MHE and EKF) and has not been applied to joint state and parameter estimation problems. In theory, the current ALS formulation should be able to accurately derive covariance estimates for use with the Dual EKF for several reasons: 1) state-space models are still utilized and are only augmented with the parameters being estimated; 2) linearization can be performed with respect to the estimated parameters; and 3) estimator gain can be readily calculated.

As a result, an attempt was made at applying ALS with the Dual EKF to the GOS process. During this test, an initial set of covariance estimates was applied to the Dual EKF to calculate an initial set of Kalman gains. These gains were utilized by the ALS algorithm to derive a set of covariance estimates, which in turn produced relatively poor estimation results when compared to the DO-based Dual EKF as the total SE was more than double. However, these results were rather preliminary and can likely be significantly improved and further developed.

Outside of the Dual EKF, work could also be conducted to extend ALS to work with the P-MHE. This specific research avenue is more challenging than the Dual EKF, as the P-MHE significantly restructures the least-squares optimization problem of the traditional MHE algorithm and does not currently have a gain function that could be used with ALS. Although MHE does not have a gain function, it is acceptable to utilize the Kalman gain as the full information problem is identical to the one step recursive technique for linear systems (Rawlings et al., 2022). If ALS could not be extended to work with the P-MHE then perhaps an alternative covariance estimation technique could be developed.

Finally, work could be done to fully build out a deployable ALS software tool in MATLAB or other coding language. Currently, ALS is a set of MATLAB files, that can be reapplied to other processes by the end user manually making modifications to the code where needed (e.g., changing N_{ALS}, modifying plot commands, manually entering dimensions of various matrices). A fully built-out app with a GUI could improve the end user experience and allow ALS to be reapplied with more ease.

7.3 Add Additional Factors into Semi-Batch Costing Formula

The presented semi-batch costing algorithm posed in this dissertation does not include the effect of capital cost on determining profitability rate. Currently, only the operating costs (i.e., feed cost, purification penalty, batch time) and potential revenue (i.e., product value) are used to derive the profitability rate. As the profitability rate is on a $/day basis, it is challenging to incorporate the effect of capital cost as this is a one-time cost. Potentially, some form of annuitization could be applied to the capital cost, thus converting this to a $/day basis for use in the profitability rate expression. Although the effect of capital cost is likely to be negligible for many bioprocesses, this would be useful for determining breakeven operating conditions and would provide a more realistic estimation of the overall process economics.

Additionally, the semi-batch costing function could be updated to handle utility costs, as this is likely to play an important role in the food industry and commodity chemical industry. Many semi-batch reactors require some form of temperature regulation, and this is not currently included in the costing function. For some processes, modification of the vessel temperature may improve product yield or minimize batch time, but at the expense of increased utility cost. Also, the mass-based purification penalty can be updated or replaced with a function that penalizes both the concentration and amount of non GOS components in the batch. Depending on the GOS product being targeted, various amounts of non GOS components (e.g., lactose) are acceptable. If these costs were included, then the DRTO would be able to account for additional information and thus improve the overall process economics.

References

Albayrak, N., & Yang, S. (2001). Production of galacto-oligosaccharides from lactose by *Aspergillus oryzae* β-Galactosidase Immobilized on cotton cloth. *Biotechnology and Bioengineering*, *77*(1), 8-19. https://doi.org/10.1002/bit.1195

Alexander, R., Campani, G., Dinh, S., & Lima, F. V. (2020). Challenges and Opportunities on Nonlinear State Estimation of Chemical and Biochemical Processes. *Processes*, *8*(11), 1462. https://doi.org/10.3390/pr8111462

Alexander, R., Dinh, S., Schultz, G., Ribeiro, M. P. A., & Lima, F. V. (2023). State and covariance estimation of a semi-batch reactor for bioprocess applications. *Computers & Chemical Engineering*, *172*, 108180. https://doi.org/10.1016/j.compchemeng.2023.108180

AspenTech. (n.d. A). *Aspen Capital Cost Estimator*. Retrieved May 02, 2023, from https://www.aspentech.com/en/products/engineering/aspen-capital-cost-estimator

AspenTech. (n.d. B). *Aspen Utilities*. Retrieved May 02, 2023, from https://www.aspentech.com/en/products/engineering/aspen-utilities

Badii, C., Bellini, P., Cenni, D., Mitolo, N., Nesi, P., Pantaleo, G., & Soderi, M. (2020). Industry 4.0 Synoptics Controlled by IoT Applications in Node-Red. *2020 International Conferences on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData) and IEEE Congress on Cybermatics (Cybermatics)*, 54-61. https://doi.org/10.1109/iThings-GreenCom-CPSCom-SmartData-Cybermatics50389.2020.00028

Bayes, T. (1763). LII. An essay towards solving a problem in the doctrine of chances. By the late Rev. Mr. Bayes, F. R. S. communicated by Mr. Price, in a letter to John Canton, A. M. F. R. S. *Philosophical Transactions of the Royal Society of London*, *53*, 370-418. https://doi.org/10.1098/rstl.1763.0053

Bélanger, P. R. (1974). Estimation of noise covariances matrices for a linear time-varying stochastic process. *Automatica*, *10*(3), 267-275. https://doi.org/10.1016/0005-1098(74)90037-5

Bellantoni, J. F., & Dodge, K. W. (1967). A square root formulation of the Kalman-Schmidt filter. *American Institute of Aeronautics and Astronautics*, *5*(7), 1309-1314. https://doi.org/10.2514/3.4189

Bellini, P., Cenni, D., Mitolo, N., Nesi, P., Pantaleo, G., & Soderi, M. (2022). High level control of chemical plant by industry 4.0 solutions. *Journal of Industrial Information Integration*, 26, 100276. https://doi.org/10.1016/j.jii.2021.100276

Bolognani, S., Tubiana, L., & Zigliotto, M. (2003). Extended Kalman filter tuning in sensorless PMSM drives. *IEEE Transactions on Industry Applications*, *39*(6), 1741-1747. https://doi.org/10.1109/TIA.2003.818991

Catenza, K. F., & Donkor, K. K. (2021). Recent approaches for the quantitative analysis of functional oligosaccharides used in the food industry: A review. *Food Chemisty*, 355, 129416. https://doi.org/10.1016/j.foodchem.2021.129416

Câmara, M. M., Quelhas, A. D., & Pinto, J. C. (2016). Performance Evaluation of Real Industrial RTO Systems. *Processes*, *4*(4), 44. https://doi.org/10.3390/pr4040044

Chemstations. (n.d.). *Performing Chemical Process Optimization in CHEMCAD*. Retrieved May 02, 2023, from https://www.chemstations.com/Knowledge_Center/Performing_Chemical_Process_Optimization_in_CHEMCAD/

Chen, S. (2017). *Comparison of Batch Versus Continuous Process in the Pharmaceutical Industry Based on Safety Consideration* [Thesis, Texas A&M University]. https://core.ac.uk/download/pdf/147255497.pdf

CoinNews Media Group Company. (n.d.). *US Inflation Calculator*. Retrieved May 30, 2023, from https://www.usinflationcalculator.com/

Córdova, A., Astudillo, C., Guerrero, C., Vera, C., & Illanes, A. (2016). Assessment of the fouling mechanics of an ultrafiltration membrane bioreactor during synthesis of galacto-oligosaccharides: Effect of the operational variables. *Desalination*, *393*, 79-89. https://doi.org/10.1016/j.desal.2015.12.020

DFE Pharma. (2006). *Lactose: Some basic properties and characteristics*. (Technical Report). DFE Pharma.

Dias, L. G., Veloso, A. C. A., Correia, D. M., Rocha, O., Torres, D., Rocha, I., Rodrigues, L. R., & Peres, A. M. (2009). UV spectrophotometry method for the monitoring of galacto-oligosaccharides production. *Food Chemistry*, *113*(1), 246-252. https://doi.org/10.1016/j.foodchem.2008.06.072

Duník, J., Kost, O., Straka, O., & Blasch, E. (2018). State and measurement noise in positioning and tracking: Covariance matrices Estimation and Gaussianity Assessment. *2018 IEEE/ION Position, Location and Navigation Symposium (PLANS)*, 1326-1335. https://doi.org/10.1109/PLANS.2018.8373523

Duník, J., Kost, O., Straka, O., & Blasch, E. (2020). Covariance Estimation and Gaussianity Assessment for State and Measurement Noise. *Journal of Guidance, Control, and Dynamics*, *43*(1), 132-139. https://doi.org/10.2514/1.G004348

Duník, J., Straka, O., Kost, O., & Havlík, J. (2017). Noise covariance matrices in state-space models: A survey and comparison of estimation methods–Part 1. *International Journal of Adaptive Control and Signal Processing*, *31*(11), 1505-1543. https://doi.org/10.1002/acs.2783

Eaton, J. W. (2012). GNU Octave and reproducible research. *Journal of Process Control*, *22*(8), 1433-1438. https://doi.org/10.1016/j.jprocont.2012.04.006

Eaton, J. W. (n.d.). *GNU Octave*. GNU Octave. Retrieved May 18, 2023, from https://www.octave.org/

Fogler, H. S. (2016). *Elements of Chemical Reaction Engineering* (5th ed.). Pearson Education. ISBN-13: 978-0-13-388751-8

Frenzel, M., Zerge, K., Clawin-Rädecker, I., & Lorenzen, P. C. (2015). Comparison of the galacto-oligosaccharide forming activity of different β-galactosidases. *LWT – Food Science and Technology*, *60*(2), 1068-1071. https://doi.org/10.1016/j.lwt.2014.10.064

Future Market Insights. (2020, February). *Galcto Oligosaccharide (GOS) Market*. https://www.futuremarketinsights.com/reports/galacto-oligosaccharide-gos-market

Gao, S., Wu, X., Luan, L., Wang, J., & Wang, G. (2018). PSO optimal control of model-free adaptive control for PVC polymerization process. *International Journal of Automation and Computing*, *15*, 482-491. https://doi.org/10.1007/s11633-016-0973-7

Google. (2023, May 18) *Google Trends*. https://trends.google.com/trends/explore?date=today%205-y&geo=US&q=%2Fm%2F0d06y,%2Fm%2F053_x&hl=en-US

Hart, P. W., & Sommerfeld, J. T. (1997). Cost Estimation of Specialty Chemicals From Laboratory-Scale Prices. *Cost Engineering*, *39*(3), 31-35.

Haseltine, E. L., & Rawlings, J. B. (2005). Critical Evaluation of Extended Kalman Filtering and Moving-Horizon Estimation. *Industrial & Engineering Chemistry Research*, *44*(8), 2451-2460. https://doi.org/10.1021/ie034308l

He, X., & Lima, F. V. (2020). A Modified SQP-based Model Predictive Control Algorithm: Application to Supercritical Coal-fired Power Plant Cycling. *Industrial & Engineering Chemistry Research*, *59*, 15671-15681. https://doi.org/10.1021/acs.iecr.0c01843

Heffes, H. (1966). The effect of erroneous models on the Kalman filter response. *IEEE Transactions on Automatic Control*, *11*(3), 541-543. https://doi.org/10.1109/TAC.1966.1098392

Jan. (2021, Aug 4). *Efficient algorithm for a duplication matrix* [Online forum post]. MATLAB Answers. https://www.mathworks.com/matlabcentral/answers/473737-efficient-algorithm-for-a-duplication-matrix?s_tid=srchtitle_duplication%20matrix_10

Julier, S. J., & Uhlmann, J. K. (2004). Unscented Filtering and Nonlinear Estimation. *Proceedings of the IEEE*, *92*(3), 401-422. https://doi.org/10.1109/JPROC.2003.823141

Kalman, R. E. (1960). A New Approach to Linear Filtering and Prediction Problems. *Journal of Basic Engineering*, *82*(1), 35-45. https://doi.org/10.1115/1.3662552

Kashyap, R. L. (1970). Maximum likelihood identification of stochastic linear systems. *IEEE Transactions on Automatic Control*, *15*(1), 25-34. https://doi.org/10.1109/TAC.1970.1099344

Kim, R. (2021). *Dynamic Optimization Algorithms for Baseload Power Plant Cycling under Variable Renewable Energy* [Doctoral dissertation, West Virginia University]. The Research Repository at WVU. https://researchrepository.wvu.edu/etd/8101/

Kim, R., & Lima, F. V (2020). A Tchebycheff-based multi-objective combined with a PSO–SQP dynamic real-time optimization framework for cycling energy systems. *Chemical Engineering Research and Design*, *156*, 180-194. https://doi.org/10.1016/j.cherd.2020.01.020

Kim, R., & Lima, F. V. (2023). Nonlinear multiobjective and dynamic real-time predictive optimization for optimal operation of baseload power plants under variable renewable energy. *Optimal Control Applications and Methods*, *44*(2), 789-829. https://doi.org/10.1002/oca.2852

Kolås, S., Foss, B. A., & Schei, T. S. (2009). Constrained nonlinear state estimation based on the UKF approach. *Computers & Chemical Engineering*, *33*(8), 1386-1401. https://doi.org/10.1016/j.compchemeng.2009.01.012

Krishnamoorthy, D., Foss, B., & Skogestad, S. (2018). Steady-State Real-Time Optimization using Transient Measurements. *Computers & Chemical Engineering*, *115*, 34-45. https://doi.org/10.1016/j.compchemeng.2018.03.021

Kummer, A., Varga, T., & Nagy, L. (2020). Semi-batch reactor control with NMPC avoiding thermal runaway. *Computers & Chemical Engineering*, *134*, 106694. https://doi.org/10.1016/j.compchemeng.2019.106694

Kühl, P., Diehl, M., Kraus, T., Schlöder, J. P., & Bock, H. G. (2011). A real-time algorithm for moving horizon state and parameter estimation. *Computers & Chemical Engineering*, *35*(1), 71-83. https://doi.org/10.1016/j.compchemeng.2010.07.012

Kulikov, G. Y., & Kulikova, M. V. (2019). Numerical robustness of extended Kalman filtering based state estimation in ill-conditioned continuous-discrete nonlinear stochastic chemical systems. *International Journal of Robust and Nonlinear Control*, *29*(5), 1377-1395. https://doi.org/10.1002/rnc.4440

Lamsal, B. P. (2012). Production, health aspects and potential food uses of dairy prebiotic galactooligosaccharides. *Journal of the Science of Food and Agriculture*, *92*(10), 2020-2028. https://doi.org/10.1002/jsfa.5712

Larsen, R. D. (1985). Box-and-whisker plots. *Journal of Chemical Education*, *62*(4), 302-305. https://doi.org/10.1021/ed062p302

Lee, S. L (2017, May 17). *Modernizing the Way Drugs are Made: A Transition to Continuous Manufacturing*. FDA. https://www.fda.gov/drugs/news-events-human-drugs/modernizing-way-drugs-are-made-transition-continuous-manufacturing

Lima, F. V., & Rawlings J. B. (2011). Nonlinear Stochastic Modeling to Improve State Estimation in Process Monitoring and Control. *AIChE Journal, 57*(4), 996-1007. https://doi.org/10.1002/aic.12308

Lima, F. V., Rajamani, M. R., Soderstrom, T. A., & Rawlings, J. B. (2013). Covariance and State Estimation of Weakly Observable Systems: Application to Polymerization Processes. *IEEE Transactions on Control Systems Technology*, *21*(4), 1249-1257. https://doi.org/10.1109/TCST.2012.2200296

Lu, M., Qiao, X., & Chen, G. (1992). Parallel computation of the modified extended kalman filter, *International Journal of Computer Mathematics*, *45*(1-2), 69-87. https://doi.org/10.1080/00207169208804119

MathWorks. (n.d.). *fmincon*. Retrieved May 03, 2023, from https://www.mathworks.com/help/optim/ug/fmincon.html

MathWorks. (n.d. B). *How GlobalSearch and MultiStart Work*. Retrieved May 09, 2023, from https://www.mathworks.com/help/gads/how-globalsearch-and-multistart-work.html

Mehra, R. K. (1970). On the identification of variances and adaptive Kalman filtering. *IEEE Transaction on Automatic Control*, *15*(2), 175-184. https://doi.org/10.1109/TAC.1970.1099422

Mirlekar, G., Li, S., & Lima, F. V. (2017). Design and Implementation of a Biologically Inspired Optimal Control Strategy for Chemical Process Control. *Industrial & Engineering Chemistry Research*, *56*(22), 6468-6479. https://doi.org/10.1021/acs.iecr.6b04753

Myers, K. A., & Tapley, B. D. (1976). Adaptive sequential estimation with unknown noise statistics. *IEEE Transactions on Automatic Control*, *21*(4), 520-523. https://doi.org/10.1109/TAC.1976.1101260

Nishimura, T. (1967). Error bounds of continuous Kalman filters and application to orbit determination problems. *IEEE Transaction on Automatic Control*, *12*(3), 268-275. https://doi.org/10.1109/TAC.1967.1098597

Odelson, B. J., Rajamani, M. R., & Rawlings, J. B. (2005). *A New Autocovariance Least-Squares Method for Estimating Nosie Covariances* (Report No. 2003-04) *Texas-Wisconsin Modeling and Control Consortium*. https://sites.engineering.ucsb.edu/~jbraw/jbrweb-archives/tech-reports/twmcc-2003-04.pdf

Petzelbauer, I., Kuhn, B., Splechtna, B., Kulbe, K. D., & Nidetzky, B. (2002). Development of an ultrahigh-temperature process for the enzymatic hydrolysis of lactose. IV. Immobilization of two thermostable β-glycosidases and optimization of a packed-bed reactor for lactose conversion. *Biotechnology and Bioengineering*, *77*(6), 619-631. https://doi.org/10.1002/bit.10110

Poli, R., Kennedy, J., & Blackwell, T. (2007). Particle swarm optimization. *Swarm Intelligence*, *1*, 33-57. https://doi.org/10.1007/s11721-007-0002-0

Purohit, J. L., & Patwardhan, S. C. (2018). Development of iterative extended Kalman filter for DAE System. *IFAC-PapersOnLine*, *51*(1), 691-696. https://doi.org/10.1016/j.ifacol.2018.05.116

Rajamani, M. R., Soderstrom, T. A., & Rawlings, J. B. (2007). *Application of a New Data-based Covariance Estimation Technique to a Nonlinear Industrial Blending Drum* (Report No. 2007-03) *Texas-Wisconsin Modeling and Control Consortium*. https://sites.engineering.ucsb.edu/~jbraw/jbrweb-archives/tech-reports/twmcc-2007-03.pdf

Rao, C. V., & Rawlings, J. B. (2002). Constrained process monitoring: Moving-horizon approach. *AIChE Journal*, *48*, 97-109. https://doi.org/10.1002/aic.690480111

Rawlings, J. B. (2019). *Autocovariance Least-Squares (ALS) Package*. Rawlingsgroup University of California, Santa Barbara. Retrieved May 18, 2023, from https://sites.engineering.ucsb.edu/~jbraw/software/als/download/index.html

Rawlings, J. B., Mayne, D. Q., & Diehl, M. M. (2022). *Model Predictive Control: Theory, Computation and Design* (2nd ed.). Nob Hill Publishing. ISBN: 978-0-9759377-5-4

Rice, R. C. (2010). *PID Tuning Guide: A Best Practices Approach to Understanding and Tuning PID Controllers* [White paper]. Control Station, Inc. https://web-material3.yokogawa.com/Yokogawa_PID_Tuning_Guide_-_csTuner.pdf

Rico-Rodriguez, F., Strani, L., Grassi, S., Lancheros, R., Serrato, J. C., & Casiraghi, E. (2021). Study of Galactooligosaccharides production from dairy waste by FTIR and chemometrics as Process Analytical Technology. *Food and Bioproducts Processing*, *126*, 113-120. https://doi.org/10.1016/j.fbp.2020.12.009

Rincón, F. D., Le Roux, G. A. C., & Lima, F. V. (2014 A). Implementation Challenges of Covariance Estimation Techniques for an Experimental Polymerization System. *2014 American Control Conference*.1743-1748. https://doi.org/10.1109/ACC.2014.6859057

Rincón, F. D., Esposito, M., Araújo, P. H. H., Lima, F. V., & Le Roux, G. A. C. (2014 B). Robust Calorimetric Estimation of Semi-Continuous and Batch Emulsion Polymerization Systems with Covariance Estimation. *Macromolecular Reaction Engineering*, *8*(6), 456-466. https://doi.org/10.1002/mren.201300151

Rincón, F. D., Esposito, M., Araújo, P. H. H., Sayer, C., & Le Roux, G. A. C. (2013). Calorimetric Estimation Employing the Unscented Kalman Filter for a Batch Emulsion Polymerization Reactor. *Macromolecular Reaction Engineering*, *7*(1), 24-35. https://doi.org/10.1002/mren.201200044

Rockwell Automation. (2016). *PlantPAx® MPC: Model Predictive Control Product* [Brochure]. Rockwell Automation. https://www.rockwellautomation.com/en-us/capabilities/process-solutions/process-optimization.html

Rodríguez, T. H., Posch, C., Pörtner, R., & Frahm, B. (2021). Dynamic parameter estimation and prediction over consecutive scales, based on moving horizon estimation: applied to an industrial cell culture seed train. *Bioprocess and Biosystems Engineering*, *44*, 793-808. https://doi.org/10.1007/s00449-020-02488-1

Rodriguez-Colinas, B., Abreu, M. A., Fernandez-Arrojo, L., Beer, R., Poveda, A., Jimenez-Barbero, J., Haltrich, D., Olmo, A. O. B., Fernandez-Lobato, M., & Plou, F.J. (2011). Production of galacto-oligosaccharides by the β-Galactosidase from *Kluyveromyces lactis*: Comparative Analysis of Permeabilized Cells versus Soluble Enzyme. *Journal of Agriculture and Food Chemistry*, *59*(19), 10477-10484. https://doi.org/10.1021/jf2022012

Sass, A., Kummer, A., & Abonyi, J. (2022). Multi-agent reinforcement learning-based exploration of optimal operation strategies of semi-batch reactors. *Computers &*

Chemical Engineering,*162*, 107819. https://doi.org/10.1016/j.compchemeng.2022.107819

Schneider, R., & Georgakis, C. (2013). How to NOT Make the Extended Kalman Filter Fail. *Industrial & Engineering Chemistry Research*, *52*, 3354-3362. https://doi.org/10.1021/ie300415d

Schultz, G., Alexander, R., Lima, F. V., Giordano, R. C., & Ribeiro, M. P. A. (2021). Kinetic modeling of the enzymatic synthesis of galacto-oligosaccharides: Describing galactobiose formation. *Food and Bioproducts Processing*, *127*, 1-13. https://doi.org/10.1016/j.fbp.2021.02.004

Scott, F., Vera, C., & Conejeros, R. (2016). Technical and Economic Analysis of Industrial Production of Lactose-Derived Prebiotics With Focus on Galacto-Oligosaccharides. In Illanes, A., Guerrero, C., Vera, C., Wilson, L., Conejeros, R., & Scott, F. (Eds.), *Lactose-Derived Prebiotics: A Process Perspective*. (pp. 261-284). Academic Press. https://doi.org/10.1016/B978-0-12-802724-0.00007-X

Segovia, V. R., & Theorin, A. (2013). History of Control: History of PLC and DCS. https://archive.control.lth.se/media/Education/DoctorateProgram/2012/HistoryOfControl/Vanessa_Alfred_report.pdf

Shin, H., Park, J., & Yang, J. (1998). Continuous production of galacto-oligosaccharides from lactose by *Bullera singularis* β-galactosidase immobilized in chitosan beads. *Process Biochemistry*, *33*(8), 787-792. https://doi.org/10.1016/S0032-9592(98)00045-4

Sigma-Aldrich. (n.d. A). *β-Galactosidase from Kluyveromyces lactis*. Retrieved May 29, 2023, from https://www.sigmaaldrich.com/US/en/product/sigma/g3665

Sigma-Aldrich. (n.d. B). *Sigma Quality Control Test Procedure: Enzymatic Assay of β-Galactosidase (EC 3.2.1.23)*. Retrieved June 29, 2023, from https://www.sigmaaldrich.com/deepweb/assets/sigmaaldrich/product/documents/288/253/g5160enz.pdf

Sigma-Aldrich. (n.d. C). *β-Galactosidase from Escherichia coli*. Retrieved May 29, 2023, from https://www.sigmaaldrich.com/US/en/product/sigma/g5635

Sigma-Aldrich. (n.d. D). *Cellulase from Aspergillus niger*. Retrieved May 29, 2023, from https://www.sigmaaldrich.com/US/en/product/sigma/c1184

Sigma-Aldrich. (n.d. E). *Pectinase from Aspergillus niger*. Retrieved May 29, 2023, from https://www.sigmaaldrich.com/US/en/product/sigma/p4716

Sigma-Aldrich. (n.d. F). *Pectinase from Rhizopus sp*. Retrieved May 29, 2023, from https://www.sigmaaldrich.com/US/en/product/sigma/p2401

Sigma-Aldrich. (n.d. G). *Protease from bovine pancreas*. Retrieved May 29, 2023, from https://www.sigmaaldrich.com/US/en/product/sigma/p4630

Sigma-Aldrich. (n.d. H). *Protease from Streptomyces griseus*. Retrieved May 29, 2023, from https://www.sigmaaldrich.com/US/en/product/sigma/p8811

Sigma-Aldrich. (n.d. I). *α-Amylase from porcine pancreas*. Retrieved May 29, 2023, from https://www.sigmaaldrich.com/US/en/product/sigma/a3176

Sigma-Aldrich. (n.d. J). *α-Amylase from Aspergillus oryzae*. Retrieved May 29, 2023, from https://www.sigmaaldrich.com/US/en/product/sigma/a9857

Sigma-Aldrich. (n.d. K). *α-Amylase from human saliva*. Retrieved May 29, 2023, from https://www.sigmaaldrich.com/US/en/product/sigma/a0521

Sigma-Aldrich. (n.d. L). *α-Amylase from Bacillus sp*. Retrieved May 29, 2023, from https://www.sigmaaldrich.com/US/en/product/sigma/a6814

Sigma-Aldrich. (n.d. M). *Bromelain from pineapple stem*. Retrieved May 29, 2023, from https://www.sigmaaldrich.com/US/en/product/sigma/b4882

Sigma-Aldrich. (n.d. N). *Ficin from fig tree latex*. Retrieved May 29, 2023, from https://www.sigmaaldrich.com/US/en/product/sigma/f6008

Sigma-Aldrich. (n.d. O). *Product Specification*. Retrieved May 29, 2023, from https://www.sigmaaldrich.com/specification-sheets/276/736/F6008-BULK SIGMA .pdf

Sigma-Aldrich. (n.d. P). *Catalase from Aspergillus niger*. Retrieved May 29, 2023, from https://www.sigmaaldrich.com/US/en/product/sigma/c3515

Sigma-Aldrich. (n.d. Q). *Lipase from Aspergillus niger*. Retrieved May 29, 2023, from https://www.sigmaaldrich.com/US/en/product/sigma/62301

Sigma-Aldrich. (n.d. R). *Glucose Oxidase from Aspergillus niger*. Retrieved May 29, 2023, from https://www.sigmaaldrich.com/US/en/product/sigma/g7141

Sigma-Aldrich. (n.d. S). *Alcohol Dehydrogenase from Saccharomyces cerevisiae*. Retrieved May 29, 2023, from https://www.sigmaaldrich.com/US/en/product/sigma/a7011

Sigma-Aldrich. (n.d. T). *Glyoxalase I from Saccharomyces cerevisiae*. Retrieved May 29, 2023, from https://www.sigmaaldrich.com/US/en/product/sigma/g4252

Slavov, A. K. (2017). General Characteristics and Treatment Possibilities of Dairy Wastewater – A Review. *Food Technology and Biotechnology*, *55*(1), 14-28. https://doi.org/10.17113%2Fftb.55.01.17.4520

Sturgeon, R. (1990). Monosaccharides. In Dey, P. M. (Eds.), *Methods in Plant Biochemistry* (pp. 1-37). Academic Press Limited. ISBN 0-12-461012-9

Sun, L., Castagno, J. D., Hedengren, J. D., & Beard, R. W. (2015). Parameter Estimation for Towed Cable Systems Using Moving Horizon Estimation. *IEEE Transactions on Aerospace and Electronic Systems*, *51*(2), 1432-1446. https://doi.org/10.1109/TAES.2014.130642

Sun, X., Jin, L., & Xiong, M. (2008). Extended Kalman Filter for Estimation of Parameters in Nonlinear State-Space Models of Biochemical Networks. *PloS ONE*, *3*(11), e3758. https://doi.org/10.1371/journal.pone.0003758

Tang, I. C. (2002). *PRODUCTION OF GALACTO-OLIGOSACCHARIDES FROM WHEY LACTOSE BY -GALACTOSIDASE IMMOBILIZED ON COTTON CLOTH*. United States Department of Agriculture – Research, Education & Economics Information System. https://reeis.usda.gov/web/crisprojectpages/0193043-production-of-galacto-oligosaccharides-from-whey-lactose-by-galactosidase-immobilized-on-cotton-cloth.html

Thierry, D., Nicholson, B., & Biegler, L. (2018). A General Framework for Sensitivity Based Optimal Control and State Estimation. *Computer Aided Chemical Engineering*, *44*, 787-792. https://doi.org/10.1016/B978-0-444-64241-7.50126-9

Turton, R., Shaeiwitz, J. A., Bhattacharyya, D &., Whiting, W. (2018). *Analysis, Synthesis, and Design of Chemical Processes* (5th ed.). Pearson Education. ISBN-13: 9780137459483

Ungarala, S., Dolence, E., & Li, K. (2007). CONSTRAINED EXTENDED KALMAN FILTER FOR NONLINEAR STATE ESTIMATION. *IFAC Proceedings Volumes*, *40*(5), 63-68. https://doi.org/10.3182/20070606-3-MX-2915.00058

USDA. (2023, May 01). *Central & Western Lactose Monthly Average Mostly Prices*. https://www.ams.usda.gov/mnreports/dymalactose.pdf

Valappil, J., & Georgakis, C. (2004). Systematic estimation of state noise statistics for extended Kalman filters. *AIChE Journal*, *46*, 292-308. https://doi.org/10.1002/aic.690460209

Valero, J. I. S. (2009). *Production Of Galacto-Oligosaccharides From Lactose By Immobilized B-Galactosidase And Posterior Chromatographic Separation* [Doctoral dissertation, Ohio State University]. OhioLINK. https://etd.ohiolink.edu/apexprod/rws_olink/r/1501/10?clear=10&p10_accession_num=osu1230909589

Varshney, D., Bhushan, M., & Patwardhan, S. C. (2019). State and Parameter estimation using extended Kitanidis Kalman filter. *Journal of Process Control*, *76,* 98-111. https://doi.org/10.1016/j.jprocont.2018.11.007

Vera, C., Guerrero, C., Illanes, A., & Conejeros, R. (2013). Fed-Batch synthesis of galacto-oligosaccharides with *Aspergillus oryzae* β-galactosidase using optimal control strategy. *Biotechnology Progress*, *30*(1), 59-67. https://doi.org/10.1002/btpr.1831

Verhaegen, M., & Van Dooren, P. (1986). Numerical aspects of different Kalman filter implementations. *IEEE Transactions on Automatic Control*, *31*(10), 907-917. https://doi.org/10.1109/TAC.1986.1104128

Wächter, A., & Biegler, L. T. (2006). On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming. *Mathematical Programming, 106*, 25-57. https://doi.org/10.1007/s10107-004-0559-y

Wang, G., Wang, H., Chen, Y., Pei, X., Sun, W., Liu, L., Wang, F., Yaqoob, M. U., Tao, W., Xiao, Z., Jin, Y., Yang, S., Lin, D., & Wang, M. (2021). Optimization and comparison of the production of galactooligosaccharides using free or immobilized

Aspergillus oryzae *β*-galactosidase, followed by purification using silica gel. *Food Chemistry*, *362*, 130195. https://doi.org/10.1016/j.foodchem.2021.130195

Wilsonm, D. I., Agarwal, M., & Rippin, D. W. T. (1998). Experiences implementing the extended Kalman filter on an industrial batch reactor. *Computers & Chemical Engineering*, *22*(11), 1653-1672. https://doi.org/10.1016/S0098-1354(98)00226-9

Yadav, J. S. S., Yan, S., Pilli, S., Kumar, L., Tyagi, R. D., & Surampalli, R. Y. (2015). Chesse whey: A potential resource to transform into bioprotein, functional/nutritional proteins and bioative peptides. *Biotechnology Advances*, *33*(6), 756-774. https://doi.org/10.1016/j.biotechadv.2015.07.002

Yang, S. T. (2003). *PRODUCTION OF GALACTOOLIGOSACCHARIDES FROM WHEY LACTOSE BY IMMOBILIZED ENZYME ON COTTON CLOTH AND NANOFILTRATION.* United States Department of Agriculture – Research, Education & Economics Information System. https://portal.nifa.usda.gov/web/crisprojectpages/0193482-production-of-galactooligosaccharides-from-whey-lactose-by-immobilized-enzyme-on-cotton-cloth-and-nanofiltration.html

Yin, H., Bultema, J. B., Dijkhuizen, L., & Leeuwen, S. S. (2017). Reaction kinetics and galactooligosaccharide product profiles of the β-galactosidases from *Bacillus circulans, Kluyveromyces lactis* and *Aspergillus oryzae*. *Food Chemistry*, *225*, 230-238. https://doi.org/10.1016/j.foodchem.2017.01.030

Zarei, J., & Shokri, E. (2014). Nonlinear and Constrained State Estimation Based on the Cubature Kalman Filter. *Industrial & Engineering Chemistry Research*, *53*(10), 3938-3949. https://doi.org/10.1021/ie4020843

Zavala, V. M., & Biegler, L. T. (2009). Nonlinear Programming Strategies for State Estimation and Model Predictive Control. In Magni, L., Raimondo, D. M., Allgöwer, F. (Eds.), *Nonlinear Model Predictive Control: Towards New Challenging Applications*. (pp. 419-432). Springer. https://doi.org/10.1007/978-3-642-01094-1_33

Zhou, Y., & Tan, Y. (2009). GPU-based parallel particle swarm optimization. *2009 IEEE Congress on Evolutionary Computation Evolutionary Computation.* 1493-1500. https://doi.org/10.1109/CEC.2009.4983119