ACCELERATING CONVERGENCE OF LEAPFROGGING OPTIMIZATION – APPLICATIONS TO NONLINEAR PROCESS MODELING AND NONLINEAR MODEL PREDICTIVE CONTROL

By

ANAND GOVINDARAJAN

Bachelor of Technology in Chemical Engineering
SSN College of Engineering, Anna University
Kalavakkam, Tamil Nadu, India
2008

Master of Science in Chemical Engineering
Oklahoma State University
Stillwater, Oklahoma, USA
2011

Submitted to the Faculty of the
Graduate College of the
Oklahoma State University
in partial fulfillment of
the requirements for
the Degree of
DOCTOR OF PHILOSOPHY
July, 2014

ACCELERATING CONVERGENCE OF

LEAPFROGGING OPTIMIZATION – APPLICATIONS

TO NONLINEAR PROCESS MODELING AND

NONLINEAR MODEL PREDICTIVE CONTROL

Dissertation Approved:

Dr. R. Russell Rhinehart

Dissertation Adviser

Dr. Gary L. Foutch

Dr. Arland H. Johannes

Dr. James R. Whiteley

Dr. Anand N. Vennavelli

ACKNOWLEDGEMENTS

A journey half way across the globe to pursue chemical engineering education would not have been possible without the encouragement and support of several people. I would like to begin by thanking (Late) Dr. S. Rajasekaran who was the primordial force and guiding light during my undergraduate days. Definitely, I would have never walked this far, if I had not thankfully caught on to his feverish passion for chemical engineering.

I owe my gratitude to Dr. R. Russell Rhinehart for his support, mentorship and advice over the last three years. His patience and encouragement towards my research, leadership and teaching has been instrumental for my successful period of stay at OSU. Much of the recognition that I have received at OSU is because Dr. Rhinehart gave me the freedom to participate and contribute to the OSU Automation Society and ChE Graduate Student Association.

I thank all my committee members – Dr. A. J. Johannes, Dr. James R. Whiteley, Dr. Gary L. Foutch and Dr. Anand N. Vennavelli, for accepting to serve on my committee and their timely suggestions. Additionally, I would like to thank Dr. Whiteley for his constant support, encouragement, and interest in my career, Dr. Gary L. Foutch for his understanding and support throughout my period as Teaching Associate of the Unit Operations Lab, and Dr. Anand N. Vennavelli for being an understanding supervisor during my summer internship at Fractionation Research Inc in 2013 and for being a friend, philosopher and guide over the last two years. Very special thanks to Shelley M. Potter for her support with the Unit Operations Lab, especially for stepping during my absence in the lab when I had scheduling conflicts/travel.

I will be failing in my duty if I do not thank Michael R. Resetarits for all his help, support, encouragement, inspiration throughout my PhD and for his career advice and guidance. I would like to thank V. Raghunathan, K. Nathaganapathi, B. Rajendran, D. Venkateswarlu and Dr. K. G. Narayanan – all of them my supervisors. Each of my supervisors left a profound positive impact, teaching me several things on the way, without telling me I am being taught.

I thank – my parents Govindarajan and Jayashree and my brother Ashwin for their unflinching support all these years, our Guru Sri. Sadasiva Bhramendrar for his blessings, my grandparents Sathianathan and Jembagalakshmi for their blessings and good wishes, my cousins Badri and Sriram for leading the way and helping me pursue a PhD, my cousins Anusha, Varsha, Varun, Naveen, and Shyam for their support.

My stay in OSU would not have been as colorful, eventful and successful without my sister-in-the-journey Upasana Manimegalai Sridhar. I would also like to acknowledge her sustained and devoted efforts with generating several results of this work.

To all the elders in my family, Gopal and Vijayalakshmi, Sridharan and Kanaka, Sridhar and Malathi, Nambi and .Padma, Murali and Chella, Andal, Narayanan and Saroja, Parthasarathy and Chitra without whom I could have only dreamt about being in the position I am in today.

I owe my gratitude to Sridhar and Kalavathi for their constant support, encouragement and motivation to think and dream big, and to Ramnarayanan and Prema without whose encouragement and support I would have never completed my PhD.

Many thanks to my lab mates Anirudh and Solomon for their help and suggestions, my friends Kumar, Krishnan, Venkatesh, Krishna NG, Suresh, Preethi, Shubha, Jeet, Vijayalakshmi Sethuraman, Samyukta, Sunil, Mounika, Yuvaraj, Arjun, Krishna CB, Mohan, Montoo, Prasanna,

Nikhil, Kinnera, and Vidvath for putting up with my idiosyncrasies over the last several years and being such fabulous roommates, friends and guides.

My journey to OSU would have never been possible without the guiding efforts of Dr. Aniruddha and Vivek – I thank them for their insight and encouragement.

Last, but not the least I wish to thank Kamalhaasan, for inspiring me with his indomitable passion towards his art – I have looked up to him every time the chips were down; and Ilaiyaraaja for inspiring me every day with his music, helping me climb out of difficult situations and keeping my balance throughout.

Name: ANAND GOVINDARAJAN

Date of Degree: JULY, 2014

Title of Study: ACCELERATING CONVERGENCE OF LEAPFROGGING OPTIMIZATION – APPLICATIONS TO NONLINEAR PROCESS MODELING AND NONLINEAR MODEL PREDICTIVE CONTROL

Major Field: CHEMICAL ENGINEERING

Abstract: Conventionally used optimization methods in chemical engineering applications such as linear programming (LP), Levenberg-Marquardt and sequential quadratic programming (SQP) handle nonlinear objective function (OF) surfaces by linearizing or assuming quadratic behavior of the surfaces [1]. Process modeling and nonlinear model predictive control (NMPC) applications, however, present OF surfaces with surface aberrations such as steep slopes, discontinuities, and hard constraints which require a robust and efficient optimization method. Therefore, an optimization method that can handle surface aberrations is required.

Leapfrogging (LF) is a recently developed direct search optimization method, potentially best-in-class, which can handle surface aberrations. LF starts with a set of players (trial solutions), randomly placed in the decision variable (DV) space. The worst player (player with the worst OF value) leaps over the best player into a reflected hypervolume [2]. The leapovers continue until all the players converge. LF is robust and efficient – with minimal computation effort (compared to conventional optimization methods), it can handle the challenges posed by nonlinear OF surfaces. LF was demonstrated on over 40 test functions and several modeling and NMPC applications. Rigorous fundamental analysis of LF is required – for a finer understanding of the method, exploring opportunities for improvement and scaling LF applications to large scale systems.

This work is focused on exploring and analyzing methods to accelerate convergence of LF, demonstrating application credibility on nonlinear process modeling of steady state binary distillation and NMPC of a binary distillation column. Accelerating convergence opens the doors for using LF in large scale problems that have several hundred variables such as real time optimization and refinery planning where computational effort and time are of essence. Distillation modeling is constrained, nonlinear, and has optimum confined to a narrow region; distillation control is multivariable, interacting, nonlinear and has severe disturbances.

Completion of this work will provide new fundamental understanding of LF which is critical for creating opportunities for algorithm improvement. Demonstrating application to nonlinear process modeling and NMPC will create application credibility, reveal practicality and serve as proof of concept that LF can be an optimizer of choice for use in the process control community.

TABLE OF CONTENTS

Chapter Page

LIST OF TABLES

Table Page

LIST OF FIGURES

Figure Page

CHAPTER I

INTRODUCTION

Optimization refers to maximizing or minimizing the value of an objective function (OF) by systematically choosing values of the variables that lead to an optimal solution. In general, a single decision variable (DV) optimization problem is specified as Equation (1),

$$\max \text{ or } \min_{DV=x} \{OF=f(x)\} \quad (1)$$

subject to $x > a$

where, f(x) is the OF and x the DV. In Equation (1) the DV is subject to a constraint, $x > a$. The OF can also be constrained, or both the OF and the DV can be constrained. Based on the nature of the OF, there can be a single optimum or multiple optima. When multiple optima are present, often, there is one global optimum and one (or several) local optima. The desired attribute of an optimizer is to find the global optimum. However, a single trial may not lead to the global optimum. Several independent trials may be required.

Optimization methods can be classified into two umbrella categories – linear programming (LP) and nonlinear programming (NLP). When the OF and constraints are both linear, an LP is used to solve the optimization problem. On the other hand, if the OF and constraints are either or both

Nonlinear, an NLP method is preferred to solve the problem. NLP methods can be broadly classified into gradient-based, heuristic or direct search, dynamic programming, stochastic, genetic, particle swarm, [3-15] etc.

For solving nonlinear optimization problems, traditionally, gradient based methods have been used. However, gradient based methods suffer from several disadvantages such as

- They require continuous and differentiable surfaces
- They cannot handle surface aberrations such as cliffs, inflections and multiple optima
- They can have numerical/analytical derivatives misdirect the search away from the optimum

Direct search or heuristic methods succeed where gradient based methods fail, in that they do not require continuous or differentiable surfaces, can handle surface aberrations, and multiple optima. However, the computational burden involved in direct search methods like particle swarm, leapfrogging (LF), genetic algorithms etc. sometimes does not justify the use of expensive computation for well-behaved problems. Further, with multi-particle or multi-player searches the time taken for all the particles/players to converge is several times the time taken to cluster around the solution. Therefore, one of the aims of this work is to address the issue of slow convergence of the multi-player, direct search leapfrogging (LF) optimization technique and explore methods to accelerate convergence.

1.1 Nonlinear Process Modeling and Predictive Control

Optimization is widely used in chemical engineering applications. Applications include model predictive control (MPC), real time optimization (RTO), process design, scheduling operations, fault detection, and data reconciliation. The chemical process industry has used both LP and NLP methods, although for different applications. NLP methods are preferred for process modeling

and design applications as most modeling and design problems are inherently nonlinear, mixed-integer, discontinuous, etc. On the other hand, LP methods are most preferred for RTO and MPC, owing to their ease of understanding and implementation and computational speed.

With increasing environmental regulations, higher product quality specifications, productivity demands, changing feedstock, and quest for higher profit margins, manufacturers need to operate over a wide range of operating conditions. Conventionally, linear approximations of nonlinear process models are used for control. Using process models for control was a marked improvement from using classic methods such as proportional integral derivative which do not "understand" a process. However, increasingly, such linear models are proving inadequate for control. Often, operating conditions are close to the operating boundaries where linear models sacrifice control performance to ensure constraint free operation. Therefore, the use of nonlinear models is necessary for improving the economics of operation, improved safety and efficiency. [16-18].

There are several ways to generate nonlinear models – first principles or empirical (and within this category there are many approaches – finite impulse response, neural networks, ARMA models, etc.). While nonlinear models are better representations of a process, nonlinear OFs present and create optimization challenges such as no guaranteed solutions, constraints, ill-behaved surfaces, etc. [1, 19, 20]. To solve nonlinear models, NLP methods which are robust, capable of handling ill-behaved OF surfaces and constraints, and simple to use are required [21]. The choice of NLP method is often a trade-off between efficiency and robustness of the optimizer. Subsequent sections describe a method to evaluate optimizers, and a new potential best-in-class optimizer called Leapfrogging (LF) which has shown tremendous potential in process modeling and NMPC applications [2].

One of the specific aims of this work concerns application of LF to MPC. MPC – also known as advanced process control (APC) – has been widely used, in the chemical and refining industry, over the last three decades. The objective of MPC is to use an explicit model of the process to predict future response, and accordingly move the process towards a desired state. At every time step, MPC solves for a future sequence of manipulated variable (MV) moves to keep the process on a desired path toward the desired state. However, since the process dynamics might not permit the controller to exactly follow the path, the objective is to minimize sum of square deviations from the path. Only the first input from the calculated sequence is sent to the process, and based on the process response, the entire calculation sequence is repeated [22, 23]. Figure 1, is a schematic that illustrates the concept of MPC [24].

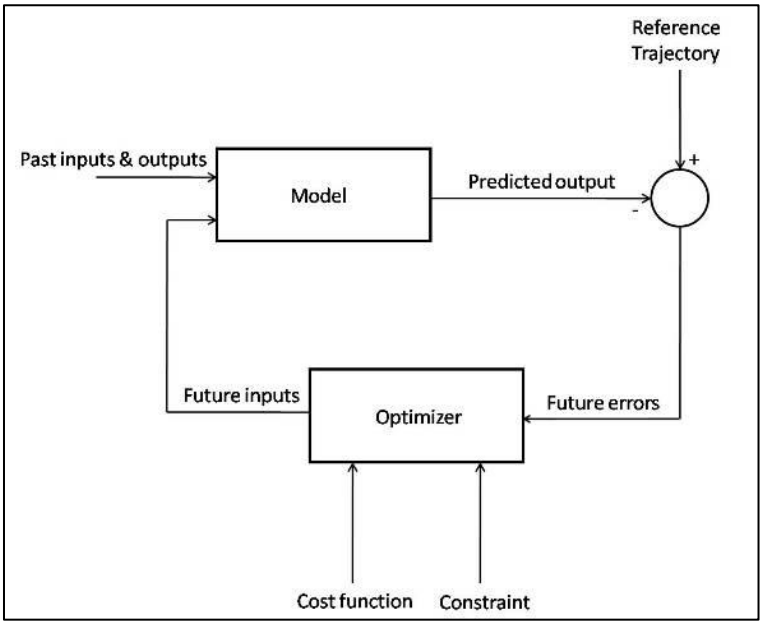


Figure 1: MPC Schematic

1.2 Conventional Optimization Methods in MPC Applications

MPC has proven benefit to the bottom line of chemical and refining process operations [20, 25, 26]. Traditionally, nonlinear OFs and constraints used in MPC are linearized and LP is used to

find optimal solution values [1, 27]. LP is fairly simple and computationally efficient, making it the *de facto* optimizer for MPC applications [1, 20]. With the advent of faster and less expensive computers in the 1990s, SQP was widely used in MPC applications [16, 22, 27-29]. SQP assumes a quadratic surface and linearizes the constraint [30-32]. Table 1 summarizes the features of LP and SQP methods.

Table 1: Summary of LP and SQP methods

Linear Programming	**Successive Quadratic Programing**
• Linearizes OF surface and constraints • Finds solution at intersection of constraints • Demonstrated application in industry **Pros** ✓ Computationally efficient ✓ Guaranteed solution ✓ Simple to understand and implement **Cons** - Solution not true optimum - Narrow operating range - Solution on constraint – extreme exterior conditioning - Jumps to new operating conditions	• Assumes quadratic surface, linearizes constraints • Finds local internal solution • Demonstrated application in industry **Pros** ✓ Computationally efficient ✓ Guaranteed solution if OF surface is well behaved ✓ Simple to understand and implement ✓ Finds interior optimum **Cons** - Uses linear constraints - Solution not true optimum - Cannot handle surface aberrations, discontinuities, inflections and multi-optima

Based on the preceding paragraphs, the following conclusions maybe drawn regarding nonlinear process modeling and nonlinear MPC (NMPC):

1. The use of nonlinear models for MPC is desired to meet modern day production and environmental demands.
2. A nonlinear programming method that can handle surface aberrations and multiple optima, and is computationally efficient is required to solve nonlinear process models and optimize control action for NMPC.
3. A proof-of-concept demonstration is required for LF as an optimizer that can handle the rigor of nonlinear process models and NMPC

Thus, the specific aims of this work are:

1. To explore and analyze methods for accelerating convergence

Accelerating convergence increases the computational efficiency of LF. To accelerate convergence, the leap-to window size is contracted. Contracting the window size speeds up LF convergence, when the players are close to the optimum by reducing the distance between the existing best solution and the relocated worst solution. Chapter II focuses exclusively on the literature concerning leapfrogging optimization

2. To evaluate results obtained by accelerating convergence

Extensive simulations are performed on standard optimization test functions and chemical engineering problems. A measure to quantify robustness and efficiency of the optimizer called – probable number of function evaluations (PNOFE), is used to compare the efficiency of the accelerated convergence modification (ACM-LF) with the original LF.

3. **To develop application credibility by**

   a. Demonstrating applicability of LF /ACM-LF to nonlinear process modeling and NMPC of a pilot scale distillation column simulation

   b. To evaluate computational performance on applications

1.3 Contributions to society

Through this work, the following contributions will be made to the existing body of work and to society at large.

1. Accelerating convergence opens the doors for using LF in large scale problems that have several hundred variables such as real time optimization and refinery planning, where computational effort and time are of essence.
2. Completion of this work will provide new fundamental understanding of LF, which is critical in creating opportunities for algorithm improvement and developing supporting mathematical analysis.
3. Demonstrating application to nonlinear process modeling and NMPC will create application credibility and proof-of-concept for practitioners. Distillation modeling based on first principles is constrained, nonlinear and has optimum confined to a narrow region. Distillation control is multivariable, interacting, nonlinear, nonstationary and typically has several disturbances. Both applications reveal practicality and serve as proof-of-concept that LF can be an optimizer of choice for use in the modeling and process control communities.

.

CHAPTER II

LITERATURE SURVEY

2.1 Leapfrogging Optimization

Chapter I provided background about the need for optimizers to be able to handle surface aberrations, multiple optima, hard constraints, etc. Recently, an optimization method called Leapfrogging (LF), has shown promise to be best-in-class for handling the above mentioned problems. Subsequent paragraphs of this chapter review the LF method, existing applications, challenges with nonlinear modeling and NMPC, optimization goodness metrics, and opportunities for improving computational efficiency and robustness.

LF is a recently developed optimization method, first published in early 2012 [2]. LF starts with a randomly located set of players (trial solutions), within the feasible DV space. At each iteration, LF relocates the worst player, by leaping across the best player into a reflected hyper volume. Equation (2) defines the leap-to position.

$$x(i)_{new} = x(i)_{current-best} - \alpha r(x(i)_{current-worst} - x(i)_{current-best}) \quad (2)$$

where, i indicates i^{th} dimension of the DV space, x(i) indicates value of the i^{th} DV dimension, current-best indicates player with the best OF value, current-worst indicates player with the worst OF value, new indicates leap-to position of the former worst, α is a scale factor that defines leap-to window size (currently a value of 1 is used) and r is a uniformly and independently distributed

random number [0,1].

Figure 2 illustrates the concept of a leapover. The contours in Figure 2 represent a simple ellipse function, with a minimum near the center. In Figure 2, the dots represent the players and the large shaded crossed circle represents the optimum. The player with the worst OF value leaps-over the player with the best OF value into a random spot in the reflected window. The leap-to position is calculated based on independent r values for each dimension and is truly random. Since r is uniformly and independently distributed, on an average it will be 0.5. Assuming that α is 1, at every leap-over the search is cut by about half the distance. Figure 3 illustrates the second leapover of LF. The worst OF spot in Figure 2 is vacated, and is indicated by a white circle with a black border in Figure 3. The new leap-to spot in Figure 2 did not find solution better than the existing best OF, nor did it find a solution worse than the previous worst OF. Therefore for the second leapover, LF searches for the current worst to leapover across the best OF. Leapovers continue until all players converge. Figure 4 is a flowchart of LF, reproduced from the original text [2].

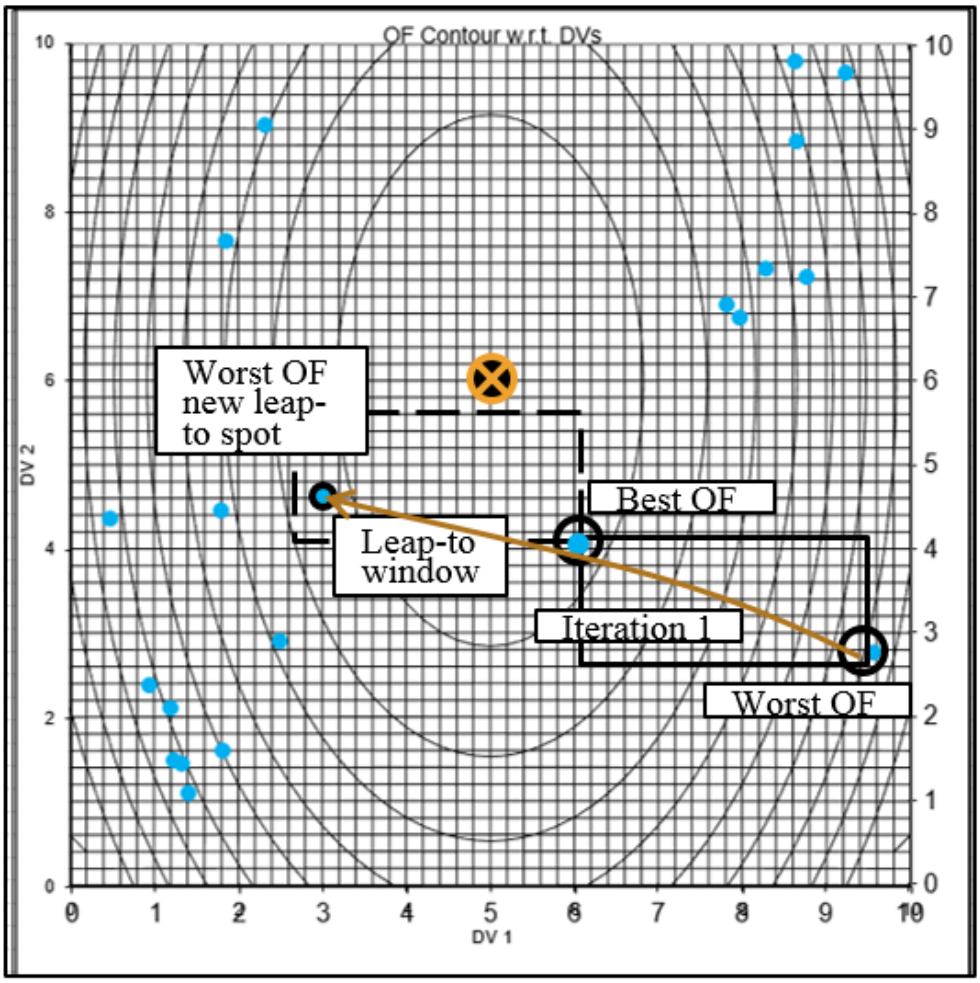


Figure 2: Leapover illustration - 1st leapover

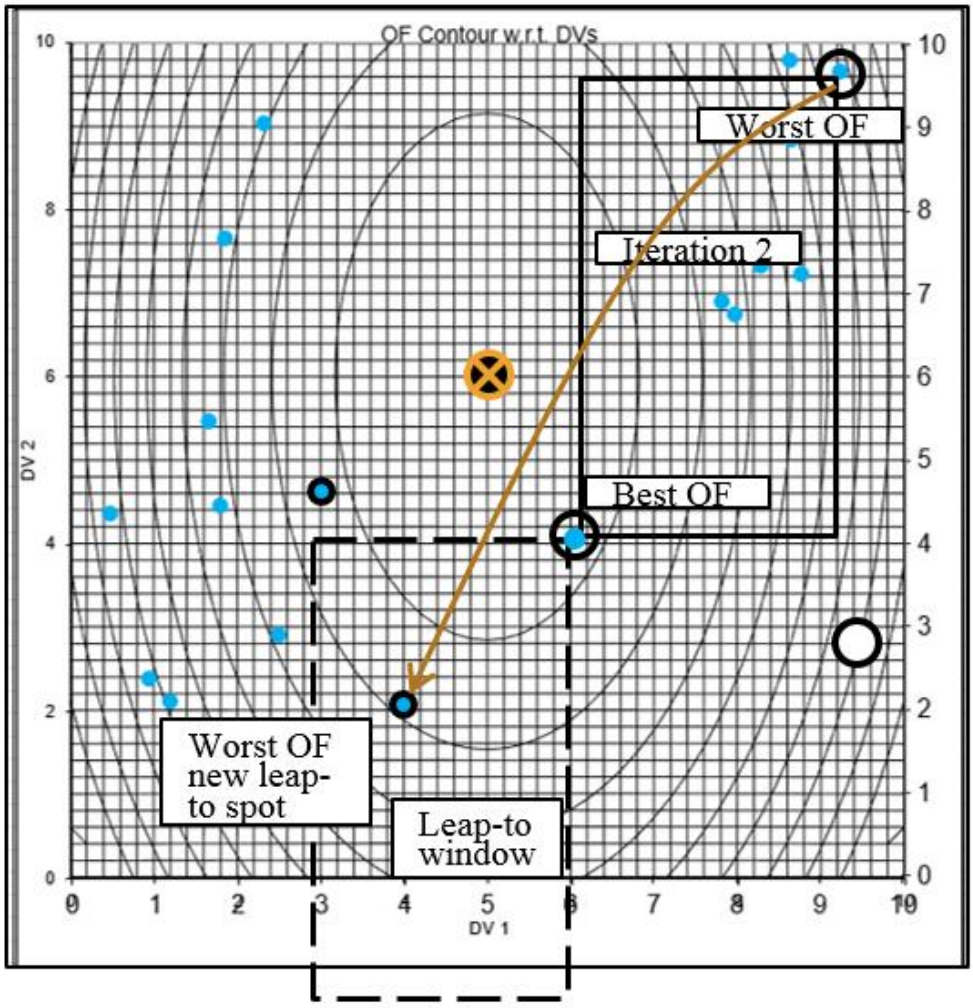


Figure 3: Leapover illustration-2nd leapover

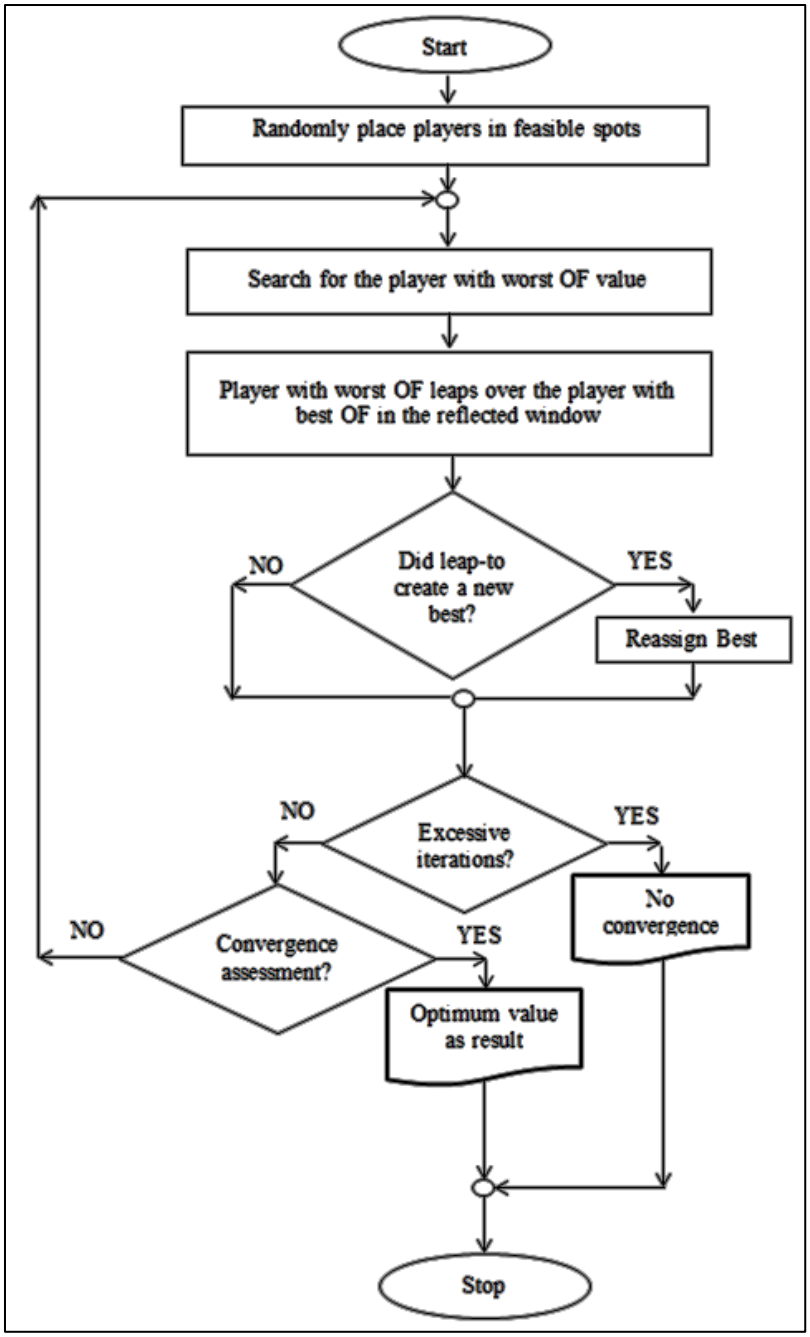


Figure 4: Leapfrogging flowchart

LF can handle both hard and soft constraints. Figure 5 illustrates how LF handles constraints. The constrained region in Figure 5 is represented by a shaded rectangular region. The shaded region is a hard constraint on the DV values i.e. a solution with a DV range bounded by the shaded region is not acceptable. The worst OF leaps over the best into the reflected DV space, but happens to

land in the constrained region. To move a player out of a constraint, LF leaps the player from the constrained spot back over the best into a new reflected window. This leap back is shown in Figure 6. Commonly, constraints are handled by other optimizers by penalizing the objective functions and therefore driving the search away from the constrained region. Penalizing objective functions needs subjective multiplier values for the penalty functions. However, LF does not require a penalty function and can directly handle hard constraints.

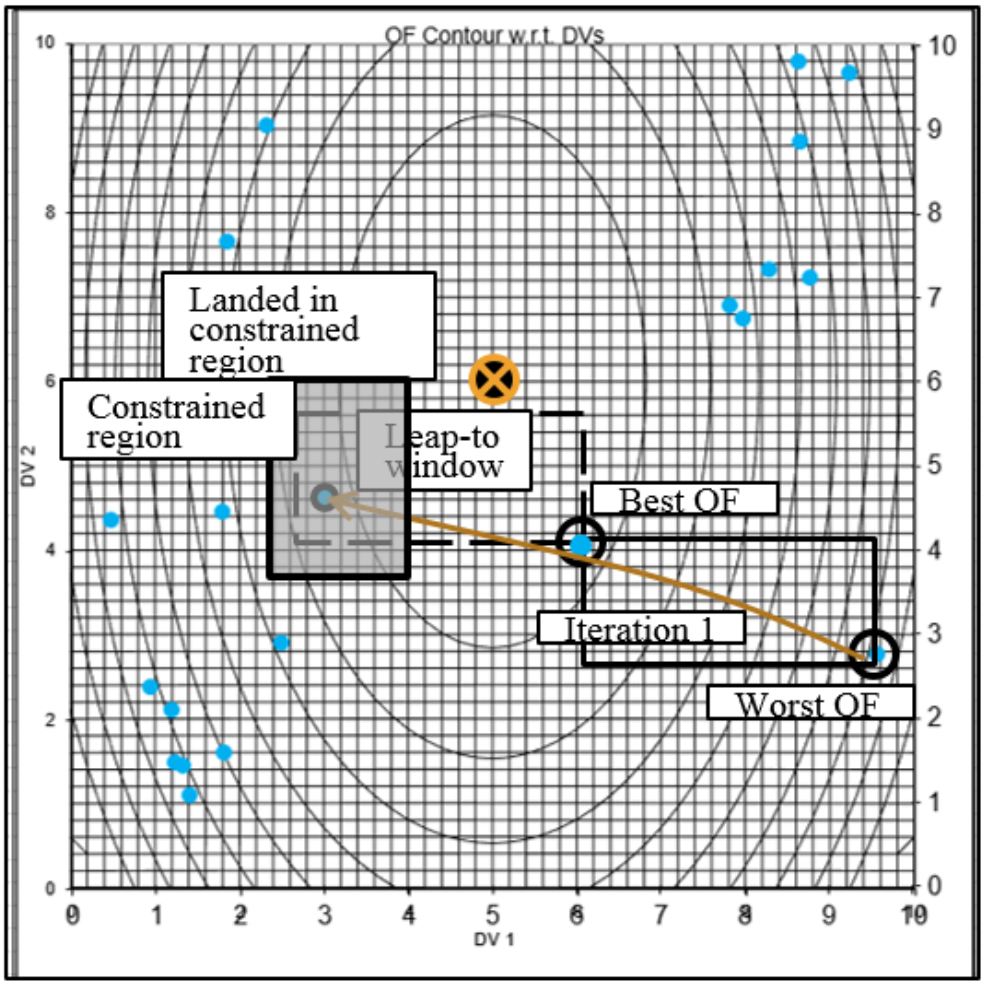


Figure 5: LF constraint handling illustration

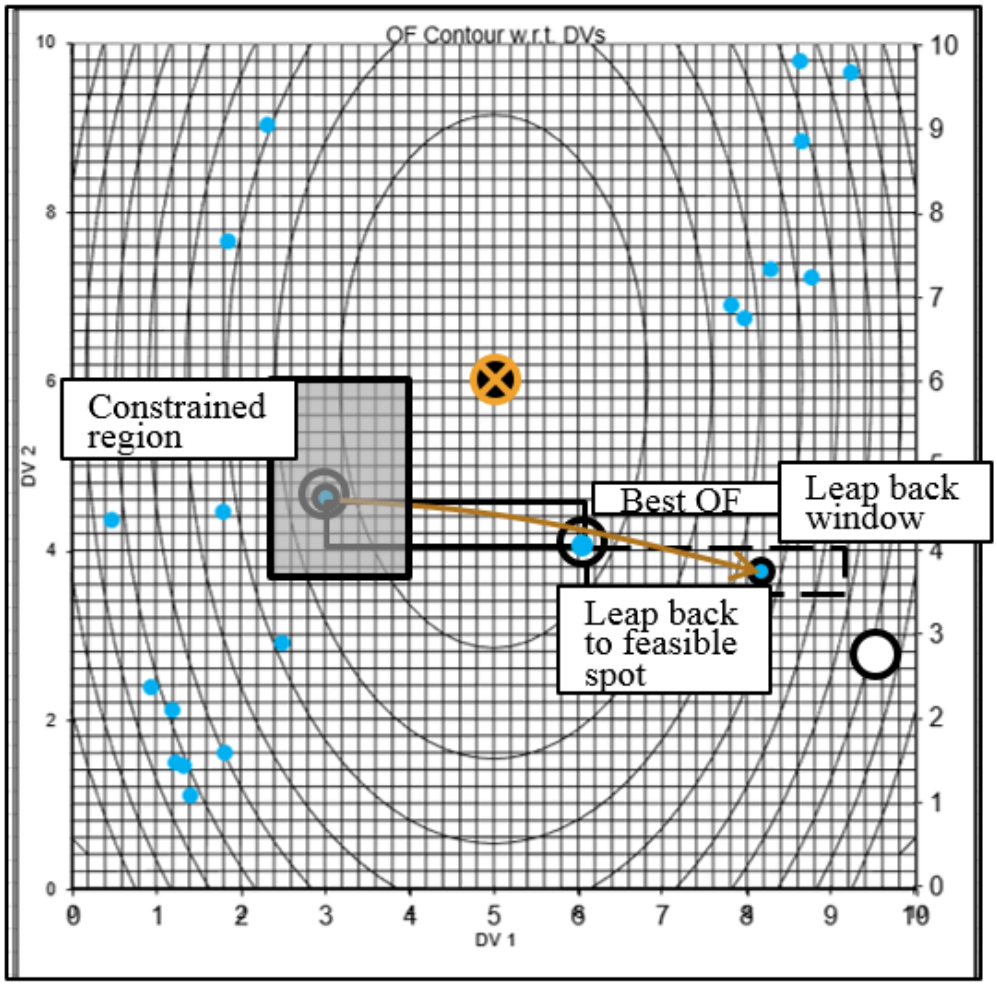


Figure 6: LF leapback to feasible spot illustration

2.2 Optimizer evaluation

A method to quantify robustness and efficiency of optimizers is required to arrive at an informed decision about the choice of optimization method. A single trial of an optimizer may not lead to the global optimum. Several independent trials may be required. If a large number of independent trials, N, are initiated, some trials will find the global, others local. With a higher N, the probability of finding the global is higher. However, with a higher N, there is additional computational burden. Computational burden is measured by number of function evaluations

(NOFE). A method to determine the number of independent starts required to find the global with a confidence c and within the fraction f of best possible solutions [33] is defined by Equation (3),

$$N = \frac{\ln(1-c)}{\ln(1-f)} \tag{3}$$

The optimizer is run several times, from random initializations and a cumulative distribution function (CDF) of all the solutions is plotted. Figure 7 shows an example to illustrate the concept defined in Equation (3). The global is located at 0.05. About 30% of the total trials, found the global. Therefore, in Equation (3), the value of f = 0.3. The user defines *c*, the confidence that in *N* trials, at least one will find the global.

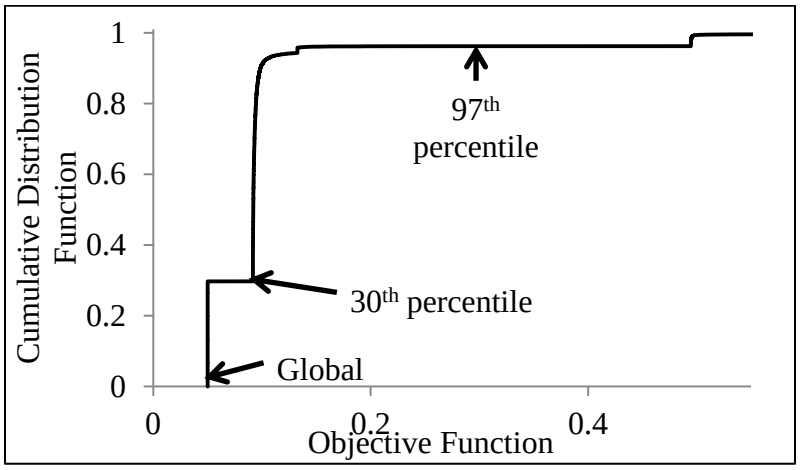


Figure 7: Cumulative distribution function of OF values

This work uses NOFE as measure of the computational burden. If ANOFE is the average NOFE per independent trial, the total NOFE over N independent starts is given by N*ANOFE. Equation (3) is used to determine N, required to be c confident that at least one of the N independent starts will find a solution within the top f fraction of possible solutions. Therefore, N*NOFE is the probable NOFE (PNOFE) required to find the global. PNOFE is used as a measure to compare the optimizers.

LF is more efficient and robust than existing optimization methods [2]. Table 2, is a summary of the effort-to-benefit (PNOFE) results for three functions using LF, Hook-Jeeves (HJ), particle swarm (PS) and modified Levenberg-Marquardt (RLM). The three test functions are representative of a variety of surface aberrations [2]. Each optimizer was run 500 times from random initializations. In Table 2, the OF value refers to the global optimum for which the CDF was determined. For each of the test functions shown in Table 2, LF has the lowest PNOFE values. In all three test cases, LF has a PNOFE lower than the other optimizers by over 50%. In the case of the function – sharp valleys with flat well, the optimization difficulty is the flat bottom to the well [2]. All optimizers except LF failed to find the global optimum for sharp valleys with flat well. Worse than not finding the global optimum, RLM encountered zero-valued gradient and hessian elements explaining the "Infinity" values in Row 4 of Table 2.

Table 2: Optimizer effort-to-benefit (PNOFE) comparison

	OF value	**Optimizer (PNOFE)**			
Function ↓		HJ	LF	PS	RLM
Bootprint with pinhole	-7.25	95,800	9,820	22,100	577,000
Sharp valleys with flat well	0.05	Infinity	2,960	Infinity	Infinity
Bootprint with constraint	0.2257	7,150	2,550	6,350	321,000

2.3 Applications

In addition to the test functions shown in Table 2, nearly 37 other functions have been demonstrated, including mixed integer and stochastic cases [2]. In publications of LF on the application front, LF has been applied to viscoelastic modeling of biological tissues, NMPC simulation and nonlinear process modeling [17, 34-39]. Table 3 summarizes applications of LF present in published literature.

Table 3: Summary of leapfrogging applications

Application	#DV's	Remarks	Application Significance
Viscoelastic modeling of soft-tissues [34, 35, 37, 38]	8	Dynamic, nonlinear, constrained, regression	Regression modeling extensively used when experimental/simulation data exists, useful in empirical modeling of process data
NMPC simulation [17, 39]	3	Dynamic, nonlinear, soft and hard constraints, single-input-multi-output, three future manipulated variable moves	Success paves way for NMPC implementation on pilot-scale process equipment
Nonlinear process modeling of a pilot-scale heat exchanger [36]	6	Dynamic, nonlinear, mixed integer	Practical application in generating dynamic, nonlinear process models useful for advanced process control
Algae bio-reactor modeling [40]	2	Stochastic, nonlinear, constrained	OF surface is stochastic creating additional challenges for the optimizer

2.4 Advantages

Based on existing results from several test cases and applications, it is clear that LF is a potential best-in-class NLP method. LF has several advantages over conventional NLP and LP methods,

- High probability of finding the global optimum
- Computationally efficient – scalability to MPC and RTO applications
- Not gradient-based – does not require continuous and differentiable surfaces
- Handles constraints – even nonlinear and hard constraints
- Does not linearize or assume quadratic OF surface
- Simple to code and execute

2.5 Improvement Opportunities

With some proven applications and advantages, LF is a potential best-in-class NLP method. However, there are significant improvement opportunities that exist, some of which authors of the original LF algorithm listed [2],

- Improved initialization – start with many individuals to increase the probability of finding the global, and then select the best subset of players for optimization. Subsequently, an improved initialization method to determine the number of initial players that increases the probability of finding the minimum with fewest PNOFE was demonstrated [41].
- Expanding and contracting leap-to window size – adjust the leap-to window based on historical trends as optimization progresses to accelerate convergence. **This work focuses on analyzing accelerated convergence by contracting the leap-to window size.**

- Combine PS and LF – start with PS when away from the global and LF near the vicinity of the global
- Adjust leap-to location – include some range near the best to draw players closer to the optimum faster
- Determine # DV's/dimension – for low dimension problems 25 players seemed the best, for higher dimensions twice or thrice the number of DV's seemed best
- Progression of work – effort to benefit analysis of increasing the number of players and reduction in PNOFE

CHAPTER III

METHODOLOGY

This chapter details the methodology used to do the following,

1. Accelerate convergence of LF
2. Develop the steady state solution of a binary distillation column
3. Develop the dynamic solution of a binary distillation column
4. Develop a NMPC application for a pilot scale distillation column

3.1 Accelerate convergence of LF

Chapter I and II have illustrated the need for a computationally efficient and robust optimizer. LF has shown tremendous promise, but there is significant scope for improving the computational efficiency. Commonly with multiplayer searches, the computational effort taken to arrive at a solution is eclipsed by the effort taken by all the players to converge. Therefore, an understanding of the mechanism of a leapover and the effort taken for convergence of all players is necessary to improve the computational efficiency of LF. At each leapover, the worst player is relocated by leaping across the best player according to Equation (4),

$$x(i)_{new} = x(i)_{current-best} - \alpha r(x(i)_{current-worst} - x(i)_{current-best}) \quad (4)$$

where, x(i) is the i^{th} decision variable (DV), r a random number between 0 and 1 – (uniformly and independently distributed UID(0,1)), and, α is the leap-to window size factor. Equation (4) maybe rearranged as,

$$x(i)_{new} = x(i)_{current\_best} - \alpha * r * d(i)_{wb} \quad (5)$$

where, $d(i)_{wb,}$ is the distance between the worst and best player of the i^{th} DV. Since r is UID(0,1), on an average its value will be 0.5. Therefore, on an average, each leapover cuts $d(i)_{wb}$ by half of α .

Figure 8 illustrates the concepts of the global attractor region and the distance of a player from the global, using 3 players. The OF in Figure 8 has two optimums and other saddle points. The point on the OF scale marked OF_1* is the global minimum while OF_2* denotes the local optimum (2^{nd} best). The region on the DV space corresponding to the hatched region on the OF space represents the global attractor area – the region where OF < OF_2*. When a player lands in the global attractor area, no player outside the area will be better, so it will draw the other players closer to it and converge at the global optimum rather than at a local optimum. Of the three players in Figure 8, Player 2 has the lowest OF value, so it is the current best. The distance between the current best and the global is designated as d_{1o}. The expected distance from global, after N leapovers will be

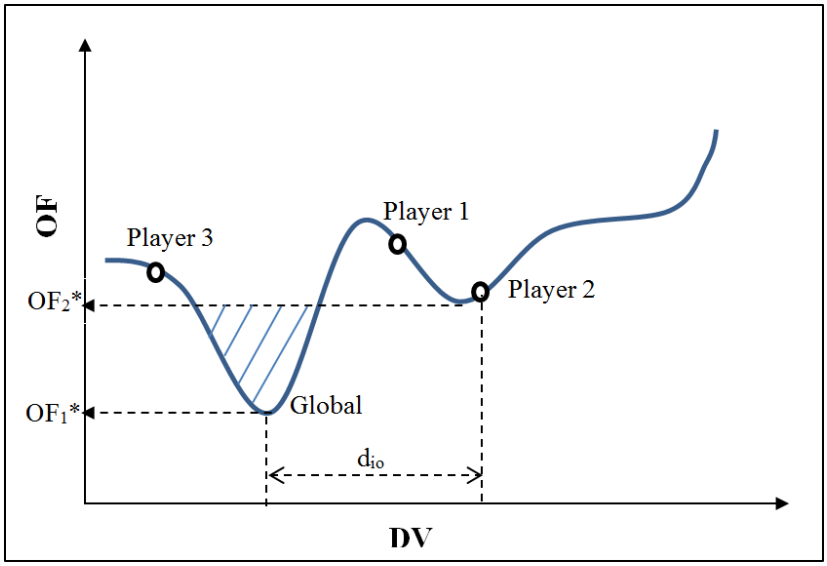


Figure 8: Concept of distance from global

$$d_{iN} = d_{io}(0.5*\alpha)^{N} \tag{6}$$

If one player is at the global, and the convergence criterion is $d_{iN} \leq \varepsilon$ then Equation (6) may be rearranged to determine the number of leapovers LF takes for a DV to converge and stop.

$$N_i = \frac{\ln\left(\frac{\varepsilon}{d_{io}}\right)}{\ln\left(0.5*\alpha\right)} \tag{7}$$

However, there has to be at least 1 leapover, and N must be an integer, therefore,

$$N_i = \text{int}\left[\frac{\ln\left(\frac{\varepsilon}{d_{io}}\right)}{\ln\left(0.5*\alpha\right)} + 1\right] \tag{8}$$

If there are M players, and one player is at the optimum, the remaining M-1 players must leap to converge. Therefore, the total number of leapovers is,

$$N_T = \sum_{i=1}^{M-1} N_i = \sum_{i=1}^{M-1} \text{int}\left[\frac{M \ln(\varepsilon) - \sum_{i=1}^{M} \ln(d_{i\text{-}wb})}{\ln(0.5*\alpha)} + 1\right] \quad (9)$$

$$= M-1 + \sum_{i=1}^{M-1} \text{int}\left[\frac{M \ln(\varepsilon) - \sum_{i=1}^{M} \ln(d_{i\text{-}wb})}{\ln(0.5*\alpha)} + 1\right]$$

In Equation (9), the term $\ln(d_{i\text{-}wb})$ is inconsequential compared to $\ln(\varepsilon)$. Therefore, the numerator of Equation (9) can be approximated as a constant "k". If Equation (9) is rearranged, the expectation is that once one player is located at the global,

$$N_T = (M - 1) + \frac{-k}{\ln(0.5*\alpha)} \quad (10)$$

From Equation (10), one may deduce that the number of leapovers to convergence once the optimum has been located is linearly proportional to the reciprocal of $-\ln(0.5*\alpha)$. Additionally, the intercept is (M-1). However, the intercept will be higher than (M-1) because the preceding analysis assumes a player is at the global, while experimental simulations take several leapovers to first find the global and then converge. Figure 9 illustrates Equation (10). The minimum number of leapovers is M-1, and with increasing α, the number of leapovers also increases. However, when $\alpha = 2.0$ the number of leapovers tends to ∞. In other words, $\alpha > 2$ will lead to divergence of the players than convergence.

Test simulations will be used to determine the correctness of this relation with respect to both ANOFE and PNOFE. The general expectation is that a lower α leads to faster convergence, i.e. lower ANOFE. However, the caveat being that faster convergence does not guarantee that LF stopped at the global optimum. A smaller window size could cause LF to converge at a local spot on the side of a hill, or at flat regions on the OF surface. Therefore, one needs to look at PNOFE which is a combined measure of computational efficiency and robustness of the optimizer.

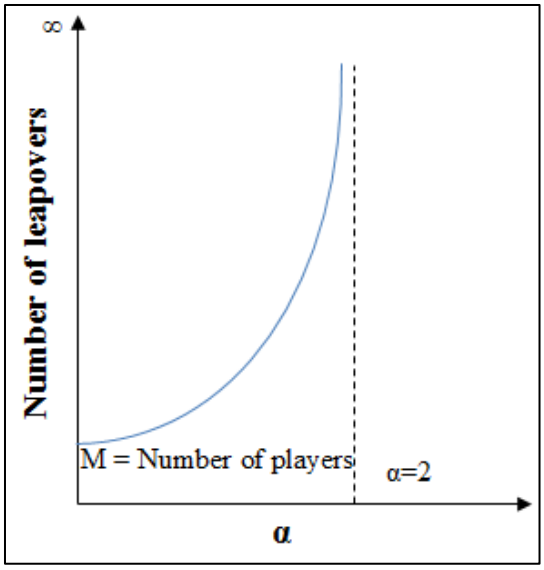


Figure 9: Number of leapovers and α

3.2 Steady State Binary Distillation Modeling

The methodology used to develop the solution to a steady state distillation column process model separating methanol-water using LF is described in this section. Determining steady state for a distillation column modeled by stage-to-stage material and energy balances requires a solution to mass, equilibrium, summation and heat equations (MESH). Therefore, the OF is constructed as a summation of the squares of the deviation of the material and energy balance equations from material and energy balance closures. In a binary distillation column, the summation and equilibrium equations also attain closure, when the material and heat balance equations are satisfied. The solution to MESH equations present the following challenges – optimum confined to a narrow region, constrained and nonlinear process model, global optimum confined to small region of DV search space and multi-scale DVs. Equations (11)-(16) are the MESH equations for a distillation column with N stages (up to 11 stages are shown in this work), a total condenser and total reboiler. For the purpose of programming, the condenser is considered stage 0 and the reboiler stage N.

Overall mass balance

$$F=D+B \tag{11}$$

Mass balance around the condenser

$$V_1 y_1 = L_o x_o + D x_d \tag{12}$$

Mass balance around the reboiler

$$L_{n-1} x_{n-1} = V_n y_n + B x_b \tag{13}$$

Mass balance over each stage

$$F_n + V_{n+1} + L_{n-1} = V_n + L_n \tag{14}$$

Component mass balance on each stage

$$F_n z_n + V_{n+1} y_{n+1} + L_{n-1} x_{n-1} = V_n y_n + L_n x_n \tag{15}$$

Energy balance on each stage

$$\Delta H_n + F_n H_{f,n} + V_{n+1} H_{n+1} + L_{n-1} h_{n-1} = V_n H_n + L_n h_n \tag{16}$$

where, F is the feed mole flow rate, D is the distillate mole flow rate, B is the bottoms mole flow rate, V is the vapor mole flow rate, L is the liquid mole flow rate, y is the vapor mole fraction, x is the liquid mole fraction, H is the vapor molar enthalpy, h is the liquid molar enthalpy, n is the n^{th} stage, and 0 is the condenser

Subsequently, the MESH equations are converted to an optimization statement. The pseudo code for the OF is as follows,

For I = 1 to NStage-1 Step 1

MBdev(I) = (Mass of (Σ Entering(I) - Σ Leaving(I))^2 '*component balance deviation on a stage*

EBdev(I) = (Enthalpy of (Σ Entering(I) - Σ Leaving(I))^2 '*energy balance deviation on a stage*

sumdev(I) = (MBdev(I) / (ECmb ^ 2)) + (EBdev(I) / (ECeb ^ 2)) *'sum of energy and*

'material balance deviation scaled using equal concern factors

OFnew = sumdev1 + sumdev(I) *'accumulate deviations across all the stages*

sumdev1 = OFnew

Next I

The material and energy balance deviations are scaled using equal concern factors. Equal concern factors are simple weighting factors (like a scalar multiplier), albeit with a physical significance. For instance, with the system of units chosen, the energy balance deviations result in significantly large numbers that are typically six orders of magnitude higher than material balance deviations. Therefore, an equal concern of 100,000 is used for the energy balance, while an equal concern of 1 is used for the material balance.

The optimizer "guesses" trial solution values for liquid mole fraction and liquid mole flows on each stage to achieve material balance and energy balance closure. For the purpose of this simulation, temperature and vapor composition of the methanol-water system are obtained as explicit functions of liquid mole fraction of methanol at atmospheric pressure, using a regression model based on literature data [42]. Further, liquid and vapor enthalpies are also obtained as explicit functions of temperature using literature data. Therefore, once the optimizer guesses a liquid mole fraction, the equilibrium temperature, vapor mole fraction, liquid and vapor enthalpies can be obtained using Equations (17)-(21) to calculate the OF.

Equilibrium temperature 'T_{eq}' (°C) as a function of liquid mole fraction 'x',

$$T_{eq}=-236.53x^5+726.38x^4-855.63x^3+490.17x^2-159.61x+99.509 \quad (17)$$

Equilibrium vapor mole fraction 'y_{eq}' as a function of liquid mole fraction 'x'

$$y_{eq}=11.21x^5-33.47x^4+37.88x^3-20.28x^2+5.65x \quad (18)$$

Equilibrium liquid mole fraction 'x_{eq}' as a function of vapor mole fraction 'y',

$$x_{eq}=-5.755y^5+12.989y^4-8.682y^3+2.352y^2-0.079y \quad (19)$$

Equilibrium liquid enthalpy 'hl_{eq}' (Btu/lbmol) as a function of equilibrium temperature 'T_{eq}'

$$hl_{eq}=-0.0167T_{eq}^3+4.0511T_{eq}^2-293.95T_{eq}+8850.2 \quad (20)$$

Equilibrium vapor enthalpy 'Hv_{eq}' (Btu/lbmol) as a function of equilibrium temperature 'T_{eq}'

$$Hv_{eq}=91.792T_{eq}+11335 \quad (21)$$

The number of DV's for this problem is 2 * NStage – 3, of which the number of liquid mole fraction variables is 1 i.e. x(1) and the number of liquid mole flow variables is Nstg-2 i.e. L(1) to L(N-1). Table 4 lists the model inputs required for simulating a steady state binary distillation column and

Table **5** lists the model outputs. While liquid mole flows of each stage in the column and the mole fraction of the top stage in the column are DVs, the other outputs listed in

Table **5** are calculated based on steady state and first principles (Equations (11)-(16)). Figure 10 is a flow chart illustrating how the players are initialized at feasible values. Based on the model inputs, the initialization procedure generates Nplayers all of which are in the feasible DV space. If any player violated a constraint, that particular player is regenerated until it satisfies all the constraints. This initialization procedure ensures that only feasible players are generated. Figure 11 illustrates how LF is used to find the solution to a steady state binary distillation column. During each iteration LF only allows feasible moves. If a player lands in a constrained region, it is leapt back out of the constrained region.

Table 4: Steady state binary distillation model inputs

Model Parameter	**Units**
Number of stages	-
Feed stage	-
Feed mole flow rate	lbmol/min
Feed mole fraction (of methanol)	-
Reflux mole flow rate	lbmol/min
Reboiler duty	Btu/min
Feed temperature	°C
Reflux mole fraction (of methanol)	-
Reboiler mole fraction (of methanol)	-

Table 5: Steady state binary distillation model outputs

Liquid mole flows of each stage	lbmol/min
Liquid mole fraction (of methanol) of each stage	-
Vapor mole flows of each stage	lbmol/min
Vapor mole fraction (of methanol) of each stage	-
Distillate mole flow	lbmol/min
Bottoms mole flow	lbmol/min
Distillate mole fraction (of methanol)	-
Bottoms mole fraction (of methanol)	-
Condenser duty	Btu/min

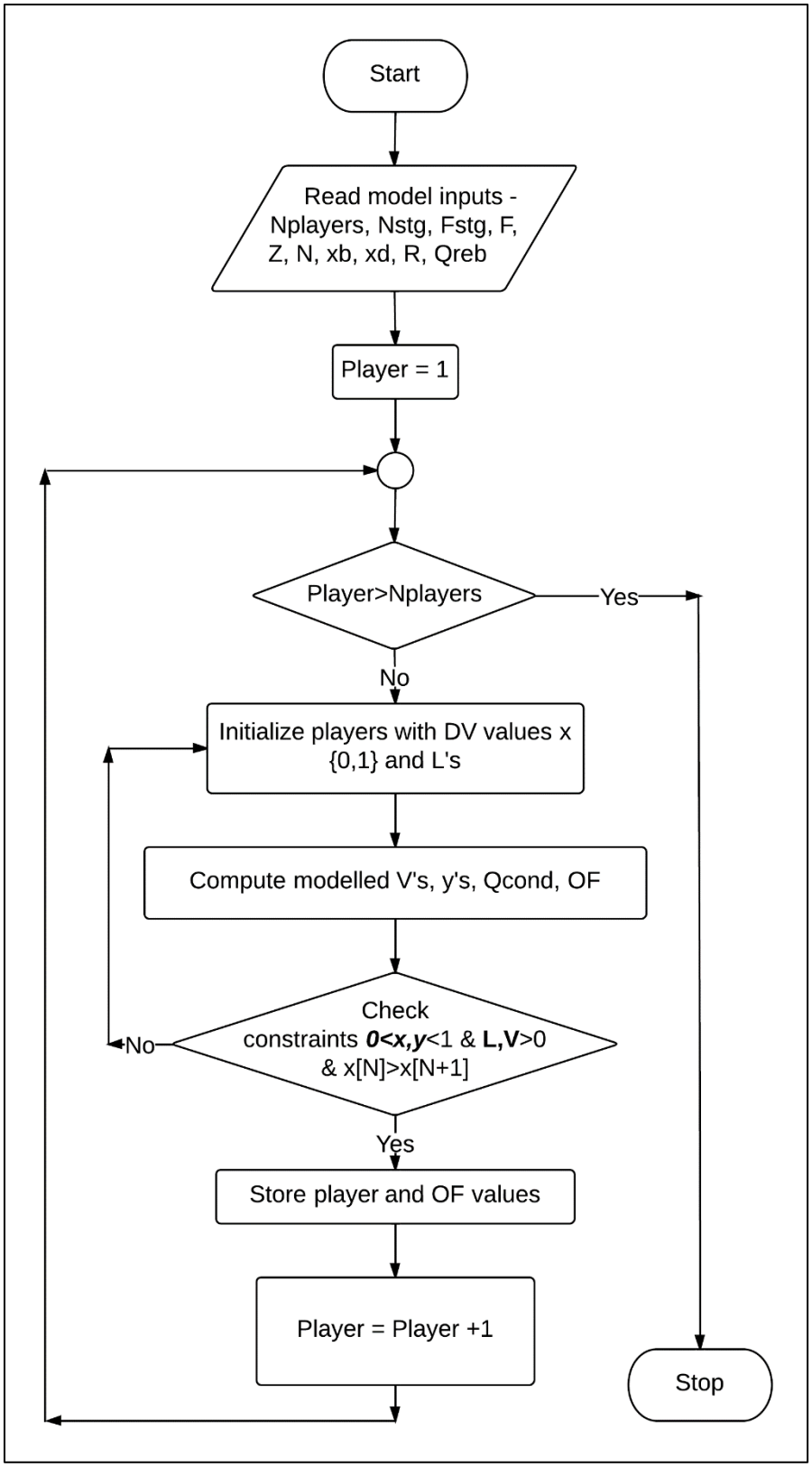


Figure 10: Initialization procedure for steady state binary distillation

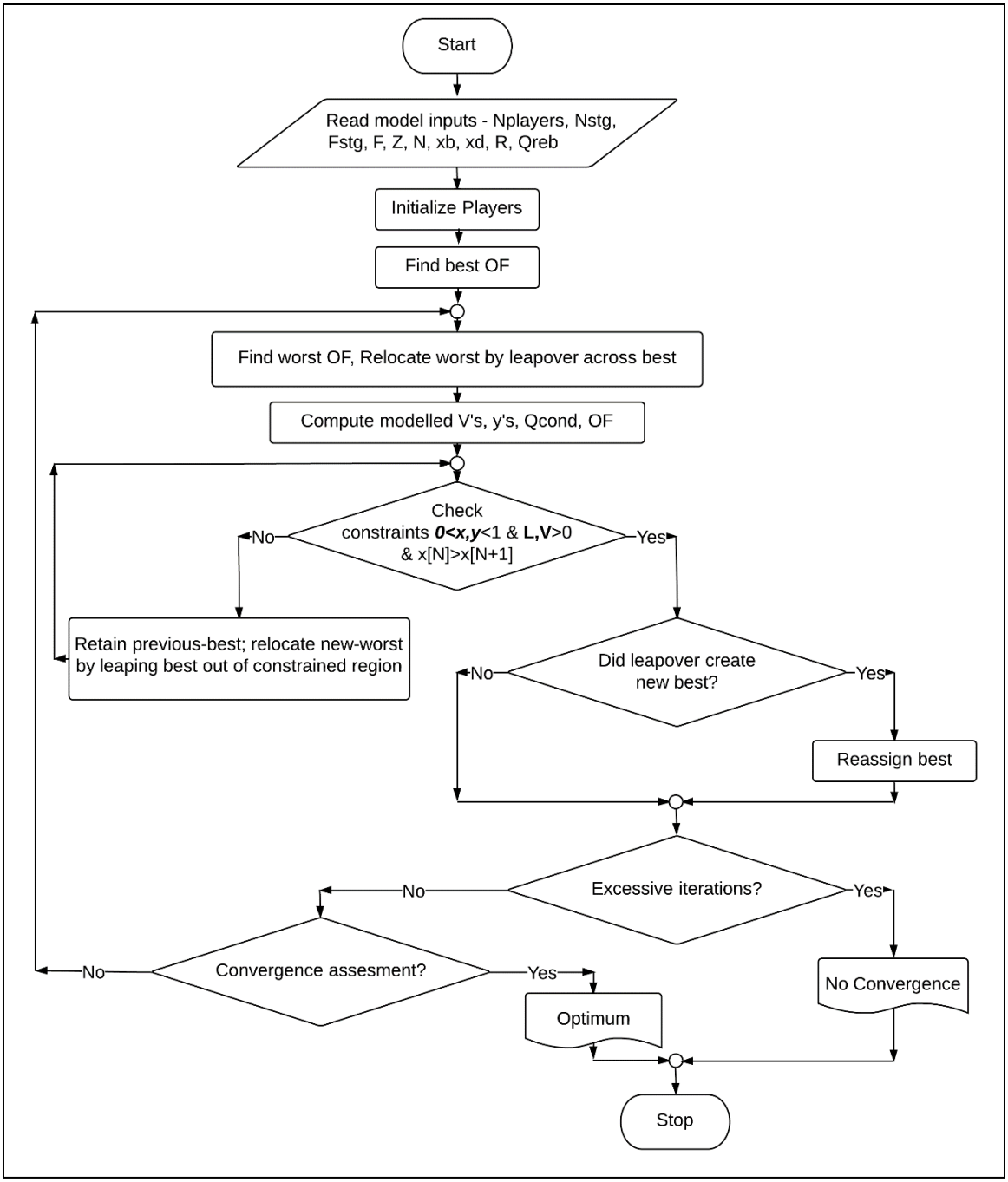


Figure 11: Optimization procedure for steady state binary distillation using LF

The steps to compute the values of the steady state (equilibrium) mole flows and mole fractions on each stage based on the model inputs (and thus the OF) is detailed below (the equation style is that of a programming language to permit ease of replication),

1. Assume a simple mixing rule to determine the latent heat of vaporization of the methanol water mixture in the reboiler (Btu/lbmol),

   $$H_{reb} = x_{reb}H_{MeOH} + (1-x_{reb})H_{H20} \quad (22)$$

2. Based on the reboiler duty (Btu/min) which is a model input and the latent heat of vaporization of methanol (Btu/lbmol) from Equation (23), the molar vapor boilup in lbmol/min is given by Equation (23),

   $$V[Nstg-1] = Q_{reb}/H_{reb} \quad (23)$$

3. Based on the molar vapor boil up and the molar liquid flow leaving the last stage in the column, the bottoms rate (lbmol/min) is calculated as,

   $$B = L[Nstg-2] - V[Nstg-1] \quad (24)$$

   Note, that the molar liquid flow on each stage is "known" because it is a quantity guessed by the optimizer.

4. From an overall material balance around the column, the distillate rate can be determined as,

   $$D = F - B \quad (25)$$

5. Based on feed composition, the equilibrium feed temperature, $T_{eq\text{-}feed}$, can be determined using Equation (17) and the equilibrium liquid molar enthalpy, $hl_{eq\text{-}feed}$, using Equation (20). However, the feed is not a saturated liquid, it is at sub-cooled condition and the enthalpy determined using Equation (20) has to be adjusted for the sensible heat change as,

   $$h_{feed} = hl_{eq\text{-}feed} - \left[\left(x_{feed} * C_{p\text{-}methanol}\right) + (1-x_{feed}) * C_{p\text{-}water}\right] * (T_{eq\text{-}feed} - T_{feed}) \quad (26)$$

6. Starting from stage 0, which is the condenser, the compositions and flows of the liquid and vapor can be determined using Equations (27)-(29). Equations (17)-(21) are used to determine the equilibrium properties on each stage. The condenser is considered to be a

total condenser, with no sub-cooling. The vapor enters the condenser as saturated vapor, condenses and leaves as a saturated liquid. Therefore the distillate composition is,

$$x_d = y[1] \quad (27)$$

7. The vapor mole flow rate on each stage is calculated using a total material balance around each stage as,

For all stages except stage 1 (28)

$$V[N] = V[N-1] - L[N-2] + L[N-1] - F[N-1]$$

For stage 1

$$V[1] = D + L[0]$$

8. The mole fraction of methanol in the vapor leaving each stage is calculated as,

For all stages except the reboiler (29)

$$y[N] = ((V[N-1]*y[N-1] - L[N-2]*x[N-2] + L[N-1]*x[N-1] - F[N-1]*x_{feed})/V[N])$$

For the reboiler, the vapor composition leaving the reboiler is determined using Equation (19) based on the reboiler composition as model input.

9. Once all the flows, compositions, temperatures, and enthalpies on each stage are calculated, the objective function can now be computed as,

$$MBdev = (V[1]*y[1] + L[1]*x[1] - V[2]*y[2] - L[0]*x[0])^2 \quad (30)$$

$$EBdev[N] = (V[N]*Hv[N] + L[N]*x[N] - L[N-1]*hl[N-1] - V[N+1]*Hv[N+1])^2$$

$$EBdev[Reb] = (L[Nstg-2]*hl[Nstg-2] + Q_{reb} - V[Nstg-1]*Hv[Nstg-1] - B*hb])^2$$

$$OF = MBdev + \frac{\sum EBdev[N]}{EC^2_{energy\text{-}balance}} + \frac{\sum EBdev[Reb]}{EC^2_{Reboiler}}$$

The energy balance deviation equations have equal concern factors, $EC_{energy\text{-}balance}$ and $EC_{Reboiler}$ to weight the deviations of the energy balance and combine the weighted

deviations to the material balance deviation. There is one material balance deviation equation and N+1 energy balance equations.

3.3 Dynamic Modeling of binary distillation

To further extend the credibility of LF, application to dynamic modeling of binary distillation will be demonstrated. A dynamic process model is required for NMPC. The steady state model developed will be used as a precursor for the development of the dynamic model. NMPC will be demonstrated on a model of the OSU, Chemical Engineering, Unit Operations Lab (UOL) pilot scale distillation unit, separating methanol-water. Pertinent details of the UOL distillation column are presented in Table 6.

Table 6: UOL distillation column details

System	Methanol-Water
Pressure	Ambient, 14.7 psia
Column internals	3.5 inch column diameter, 5 Sieve trays, with 6 inch spacing
Reboiler	Electric Bayonet Heater – maximum 4 kW, controlled with a thyristor, 0.51 ft^3
Condenser	Total condenser
Accumulator	Glass, 0.048ft^3 with overflow tube for distillate product

The main lags are due to the mixing in the reboiler and the accumulator; liquid hold up on the trays is significantly smaller than the volume of the liquid in the reboiler or accumulator. For instance, if the clear liquid height on the tray was 2 inches, the volume of liquid hold up on tray would be 0.01 ft^3 which is about 50 times smaller than the volume of the reboiler. For the sake of simplicity of modeling, this work assumes that the dynamics on the UOL pilot scale distillation unit is dominated by the dynamics in the reboiler and accumulator composition changes.

Therefore, the dynamic model has mixing dynamics on the reboiler and accumulator but steady state assumption for all the trays. Consequently, the time taken for the trays to attain steady state is inconsequential compared to the dynamics on the reboiler and accumulator. The above assumption may not be true for actual columns in operation in the industry, where tray hold up is significant and cannot be ignored. As a next step, it is required to identify the nature of the dynamics on the reboiler and accumulator. From existing data regarding the UOL distillation column (reference), it is understood that the reboiler and accumulator follow first-order dynamics. Therefore, component material balance (of methanol) around the reboiler and the accumulator are carried out, as shown in Equations (31) and (32).

$$\Delta(Vol_{accum\text{-}molar}) = V[1]*y[1]-L[0]*x[0]-D*x[0] \tag{31}$$

$$\Delta(Vol_{reb\text{-}molar}) = L[Nstg-2]*x[Nstg-2]-V[Nstg-1]*y[Nstg-1]]-B*x_b \tag{32}$$

Equations (31) and (32) may be rearranged to determine the accumulator and reboiler composition as,

$$x_{dn}= x_{do}+\left(\frac{dt}{Vol_{accum\text{-}molar}}\right)*(V[1]*(y[1]-x_{do})) \tag{33}$$

$$x_{bn}= x_{bo}+\left(\frac{dt}{Vol_{reb\text{-}molar}}\right)*(L[Nstg-2]*x[Nstg-2]-V[Nstg-1]*y[Nstg-1]]-B*x_{bo}) \tag{34}$$

where, the subscript “n” stands for the new composition, and “o” stands for the composition at the previous time step. Equations (33) and (34) are numerical approximations to first order equations. The procedure used to perform dynamic simulation is illustrated by the flowchart in Figure 12. The dynamic model is initialized with a steady state solution. At each time interval, the reboiler and accumulator compositions are updated. Subsequently, the steady state compositions and flows on each tray corresponding to the dynamic reboiler and accumulator composition is

determined. This process of updating the dynamic reboiler and accumulator compositions and finding the steady state values on the trays continue till the end of the simulation time.

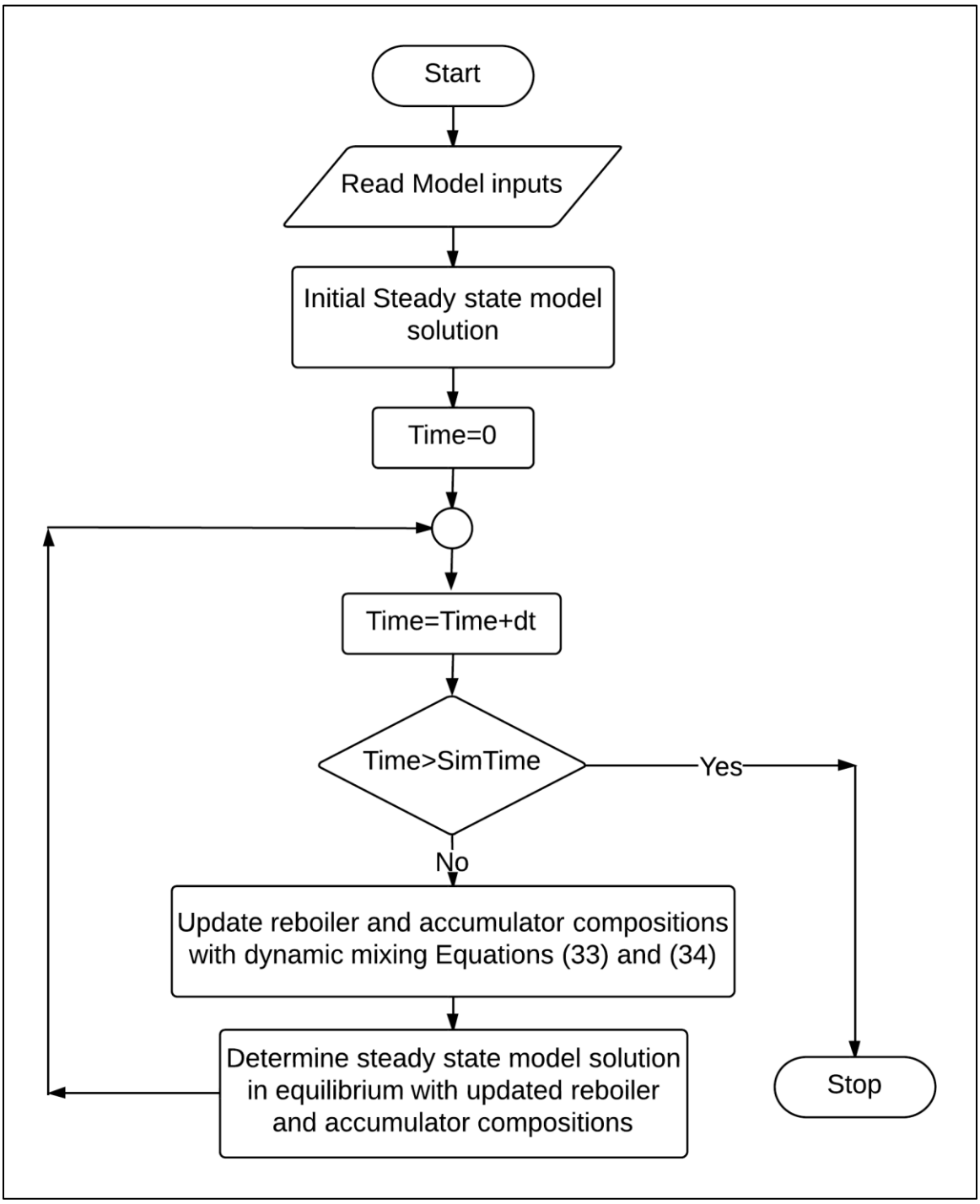


Figure 12: Dynamic simulation

3.4 Nonlinear Model Predictive Control

Model predictive control is an advanced method of process control which has been used in the process and refining industry from the 1980's. MPC requires the following elements:

1. A dynamic model that maps the relation between the manipulated variable(s) (MV) and controlled variable(s) (CV). Frequently, there are auxiliary variables such as levels, pressure drop etc. that also require an explicit relationship with the MVs. There are two dynamic models –
    a. A past-to-now (P2N) model that updates CV values, once at each controller sampling, based on the MV values of the past sample.
    b. A now-to-future (N2F) model that predicts CV values over a future time horizon based on the MV values guessed by an optimizer. The N2F model is initialized at the current sampling with the P2N model values.

    Both, the P2N and the N2F models are similar in structure (Figure 12), and only different in the purpose they serve. The steady state and dynamic models developed in Sections 3.2 and 3.3 took in excess of several minutes when used in the NMPC application. With a control interval of 1 minute, clearly the models developed in Sections 3.2 and 3.3 cannot be used for control. Therefore, to increase the speed of computation, the first principles steady state model developed in Sections 3.2 is simplified as follows,

    - The liquid flow on all stages above the feed stage is assumed to be equal to the reflux flow rate.
    - The liquid flow on the feed stage and all the stages below it are assumed to be equal to the sum of the reflux flow rate and the feed flow rate.

- The above two assumptions eliminate the need for using the liquid mole flow rates as DVs in the steady state model solution. The liquid mole fraction on the top stage of the column is the only DV.

The above assumptions permit the optimizer used for determining the future set of control actions to determine a solution in less than 20 seconds. It is important to note that, with the available computational resources the rigorous models could not be used, but in the near future, such rigorous models might become the norm rather than the exception.

Continuing with the elements of MPC,

2. An OF that is to be optimized. Commonly OF's are based on cost. For this work, the OF is the sum of squared deviations between a reference trajectory and the model predicted value based on the dynamic MV-CV relation.
3. A method to handle the mismatch between the process and the model predicted value is required. There are several methods to adjust the setpoint viz. biasing the setpoint by the process model mismatch (pmm), biasing the setpoint by the integral of the error, bias model by residual, or adjusting the model coefficients by pmm.
4. The control horizon, – the time out in the future for which a set of MV actions are determined that minimize the OF.
5. The number of MV moves and the timing of their implementation in the future horizon.
6. Reference trajectory dynamics – the manner in which the model is moved towards the biased setpoint. Based on historical data or open loop responses, the dynamics of the process can be understood and a suitable reference trajectory used. For this work, a simple first order reference trajectory that makes the model move towards the biased setpoint is used. The reference trajectory is initialized with the CV values from the P2N

model. Equations (35) and (36) are the reference trajectory for the top and bottom respectively, shown in the form of computer assignment statements.

$$\text{refxd} := \left(\frac{dt}{\tau_d}\right) * \text{xdspb} + \left(1 - \left(\frac{dt}{\tau_d}\right)\right) * \text{refxd} \tag{35}$$

$$\text{refxb} := \left(\frac{dt}{\tau_b}\right) * \text{xbspb} + \left(1 - \left(\frac{dt}{\tau_b}\right)\right) * \text{refxb} \tag{36}$$

Once all the elements listed above are available, the following steps are followed for implementing NMPC.

Step 1: An optimizer (here LF) guesses at two sets of MV moves. One set of MV moves for the reflux rate, and another for the reboiler duty. Then the model forecasts the results on all the CVs – two in this case, the top and bottom tray composition.

Step 2: The objective function is evaluated. In many MPC applications the OF is the squared deviation of the setpoint and the control variable over the future time horizon with a penalty for large MV moves. In some MPC applications, the OF is based on the squared deviation from the reference trajectory. This work uses the squared deviation of the controlled variable from the reference trajectory. The sum of squared deviations between the reference trajectory and the model values for both the CVs is the objective function. One of the CVs can be assigned a higher priority by weighting the sum of squared deviations (of that CV) with an equal concern factor (the other has a default value of 1),

$$\text{SSDxd} = \sum_{\text{Horizon=0min}}^{\text{Horizon=20min}} (\text{refxd} - \text{xdn2f})^2 \tag{37}$$

$$\text{SSDxb} = \sum_{\text{Horizon=0min}}^{\text{Horizon=20min}} (\text{refxb} - \text{xbn2f})^2$$

$$OF = \frac{SSDxd}{EC_{xd}^2} + SSDxb$$

Step 3: Repeat Steps 1 and 2 in tandem until the best OF (minimum) is achieved.

Step 4: From the best solution, implement the first MV move of each MV.

Step 5: Wait until the next controller sampling, calculate pmm, and update the dynamic P2N model. Adjust the setpoint for the model with the pmm.

Step 6: Restart Step 1.

This work demonstrates the use of LF as the optimizer for determining three future MV moves for each CV, i.e. a total of 6 MV moves (6 DVs). The NMPC simulation is developed using National Instruments' LabVIEW® software. Illustrations and details of the program developed and the human machine interface are provided in the Appendix.

Table 7 lists all the features of the NMPC used in this work. Leapfrogging with a leap-to window size factor of 0.5 was used as the optimizer in this work. The stopping criteria used for the controller-optimizer is σReboiler-MV ≤ 1 and σReflux-MV ≤ 6.67 x 10-5 i.e. when the standard deviation of the MV values of all the 60 players is below a certain threshold, LF stops.

The next chapter details the experimental testing that was carried out for demonstrating accelerated convergence, steady state and dynamic simulation, and NMPC.

Table 7: NMPC features used for binary distillation

Controlled variables	1. Top tray composition (mole fraction) 2. Bottom tray composition (mole fraction)
Manipulated variables	1. Reflux rate (lbmol/min) 2. Reboiler duty (Btu/min)
Control points	1. Tray 1 (for top tray composition) 2. Tray 5 (for bottom tray composition)
Process model mismatch handling	By biasing the setpoint with pmm i.e. ysp-bias=ysp-pmm
Controller sampling time	1 minute
Controller horizon	20 minutes
MV moves and location	3 moves for each MV, MV1: 1-10 minutes, MV2: 11-15 minutes, MV3: 16-20 minutes
Disturbances	1. Feed rate (lbmol/min) 2. Feed composition (mole fraction)
Optimizer	Leapfrogging with a leap-to window size factor (α) = 0.5, and 10 players per DV i.e. 6 DVs x 10 players = 60 players
Maximum iterations	5000
Stopping criteria	$\sigma_{\text{Reboiler-MV}} \leq 1$; $\sigma_{\text{Reflux-MV}} \leq 6.67 \times 10^{-5}$

CHAPTER IV

EXPERIMENTAL METHODS

This chapter explains the experimental methods used to test the following:

1. Accelerated convergence improvement – extensive experimental testing is used to determine the correctness of the mathematical analysis concerning accelerated convergence explained in Chapter III.
2. Steady state binary distillation modeling – the purpose of the testing is to demonstrate the ability of LF in handling a nonlinear process modeling problem, which has constraints on the DVs, interaction between the DVs and the optimum is confined to a narrow region.
3. Dynamic binary distillation modeling – to demonstrate nonlinearity of the process.
4. Nonlinear model predictive control – to show proof-of-concept of the application of LF to a multivariable control problem such as distillation that has interactions, nonlinearities and severe disturbances.

4.1 Accelerated convergence improvement

Several two DV functions which exhibit various surface difficulties are considered for conducting the analysis on the leap-to-window size factor α. Table 8 lists the 2-DV functions and a 10-DV steady state distillation column model along with the problem features. Figure 13 through Figure 20 are three dimensional views of the functions F1-F8 listed in Table 8.

Table 8: Function Description

Function	#DVs	Function Name	Minima	Function Description
F1	2	Peaks	Multiple	3 well-shaped minima
F2	2	Boot Print with Pinhole	Multiple	Up on the snow surface there is a pinhole representing the global minimum
F3	2	Goldstein-Price	Multiple	Irregular flat valley in-between steep walls
F4	2	Simple Ellipse	Single	Well behaved
F5	2	Hot & Cold Water Mixing MPC	Single	Twin objectives of hot and cold water temperatures, balanced by equal concern factors.
F6	2	1- Tray Distillation Colum	Single	1 tray distillation model with a total reboiler, objective is to close the material and energy balance
F7	2	Sharp Troughs	Multiple	3 minima, one conventional, one with gentle slope and one on a shelf
F8	2	Jupitor's Eye	Single	Twisted slot bottom to the hole containing the optimum
F9	10	Steady state 11 stage distillation column	Single	Sum total of the energy balance on each stage and the overall material balance. Equal concern factors to balance the mass balance and energy balances.

.

Figure 13 is a three dimensional view of the peaks function (F1). The peaks function has multiple optima viz. three minima and three maxima. The shape of the minima themselves are well behaved, however the presence of multiple optima has the potential to confound optimizers.

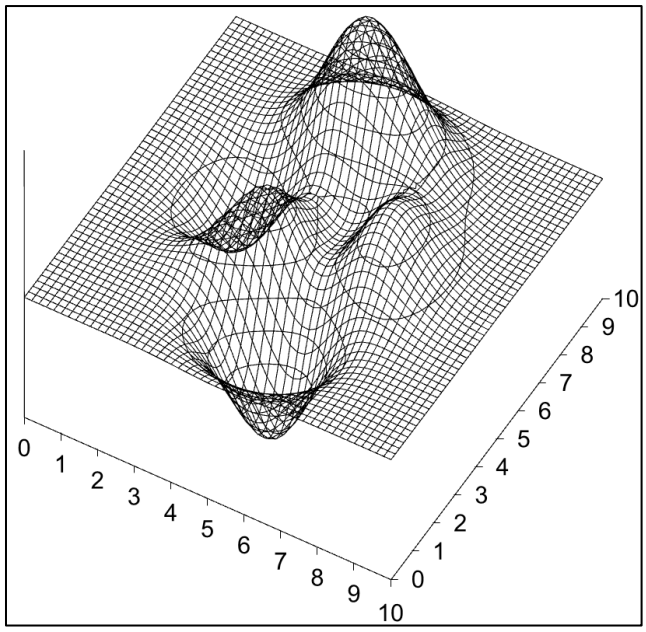


Figure 13: Peaks function (F1)

Figure 14 is the boot print with pinhole function (F2). The function resembles a boot print in snow, with a local minimum at the bottom of the boot print, surrounded by steep walls. However, the global optimum is present at the pinhole. The difficulty with F2 for optimizers is the obscure location of the global optima, with most searches leading towards the local minimum at the bottom of the boot print rather than the global minimum at the pinhole.

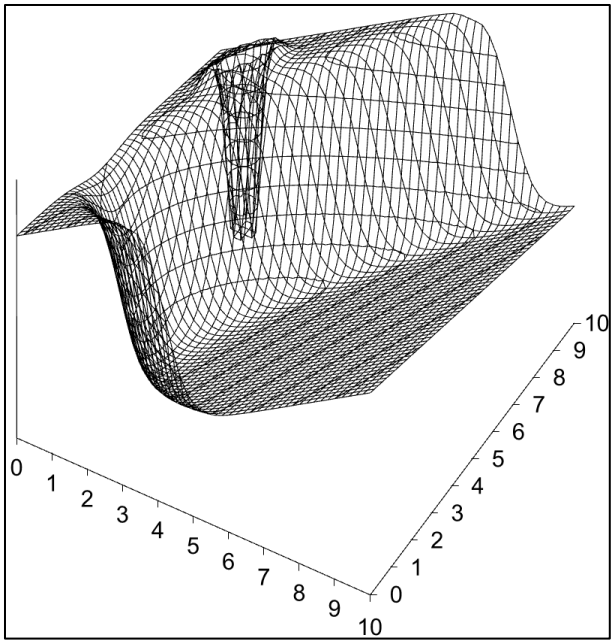


Figure 14: Bootprint with pinhole (F2)

Figure 15 is the Goldstein and Price function (F3). F3 has several local optima and one global optimum. The global optimum is located at the bottom of the gentle slope. The global optimum is surrounded by several local optima on the gentle slope, thereby confounding searches.

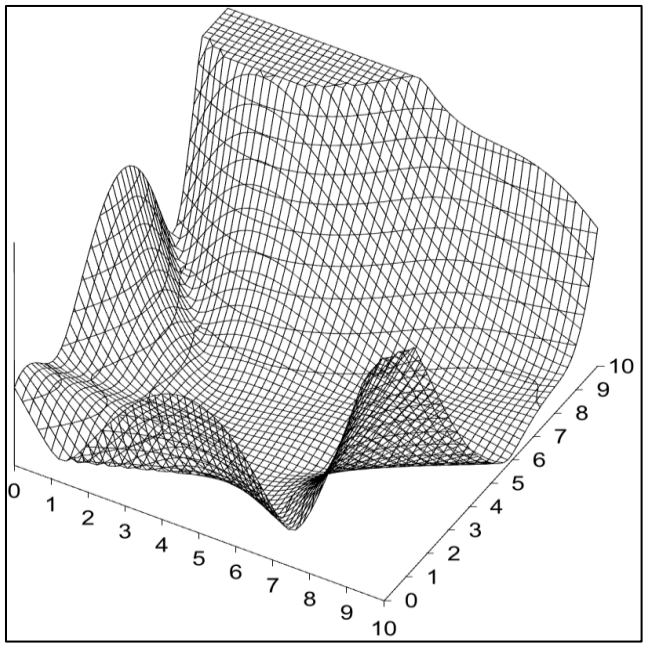


Figure 15: Goldstein and Price function (F3)

Figure 16 is a simple ellipse function (F4), with a well-behaved optimum. F4 permits analysis of modifications of LF.

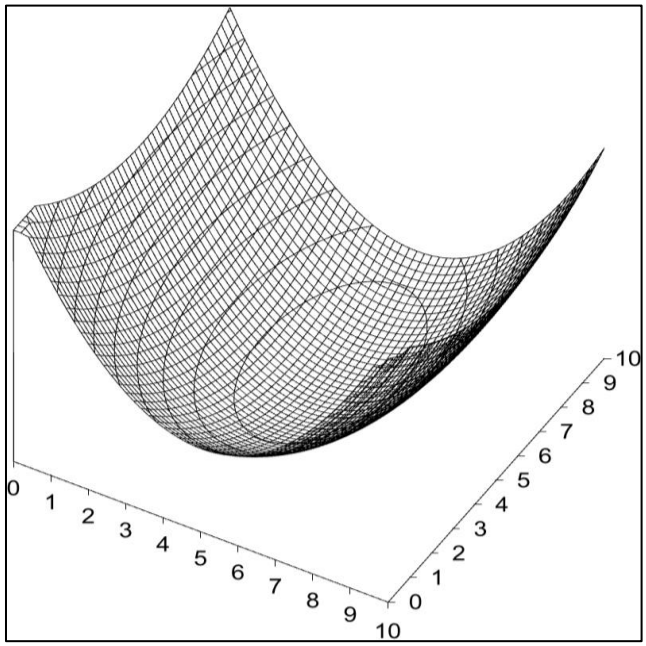


Figure 16: Ellipse function (F4)

Figure 17 is a hot and cold water mixing MPC objective function view (F5). F5 is a multi-input single output control problem which is used to control the mixed water temperature and flow rate by manipulating the hot and cold water flow rates. For optimization this has two DVs, however for control this is 2 MVs and 2 CVs. F5 was chosen to show relevant process control applications of LF and its modifications. The global is located at the bottom the gentle slope. However, F5 has some severe features – steep cliffs closer to the minimum and at the extreme of one DV.

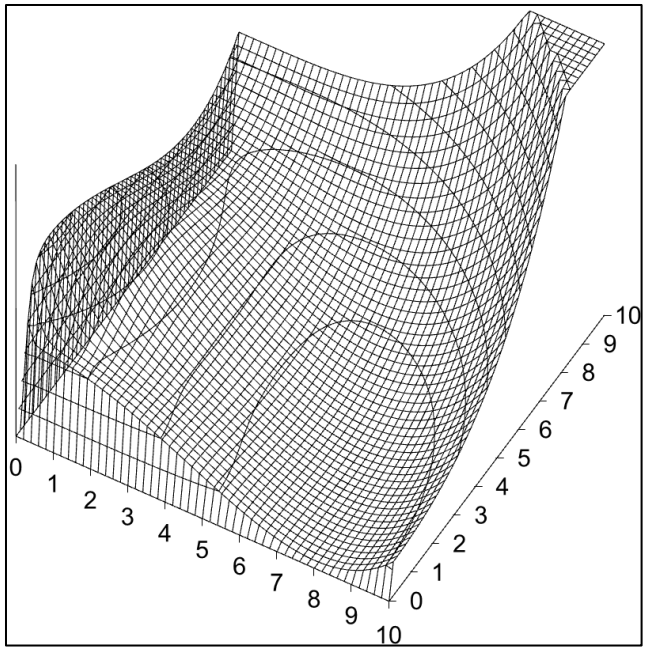


Figure 17: Hot and cold water mixing - model predictive control (F5)

Figure 18 is a one-stage distillation column (F6) with a reboiler, where the OF is formulated as the deviation of the mass and energy balance of the single stage and the reboiler. The material and energy balances are weighted by equal concern factors. F6 was chosen to reveal application.

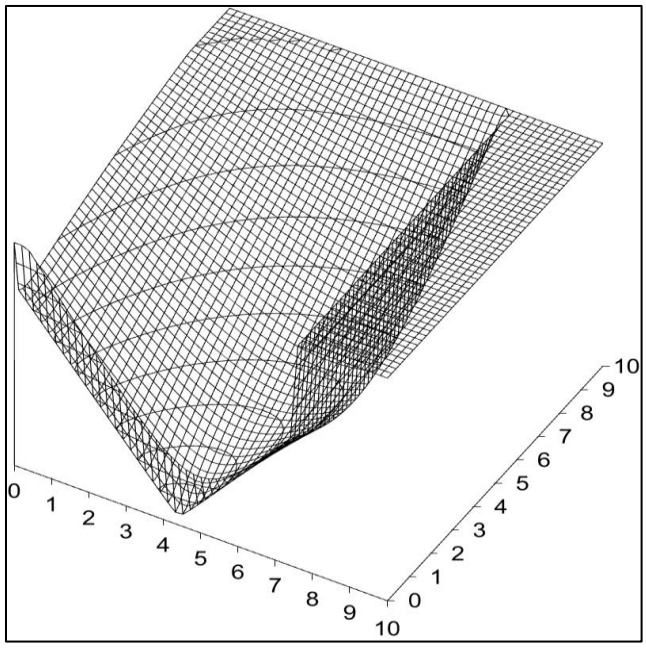


Figure 18: 1-stage distillation column (F6)

Figure 19 is a sharp troughs function with two local optima and one global optimum (F7). The global is located in the valley, and one local is located on a corner of the OF surface while the second is located on a gentle slope. F7 confounds optimizers because of the sharp valleys and the presence of multiple optima.

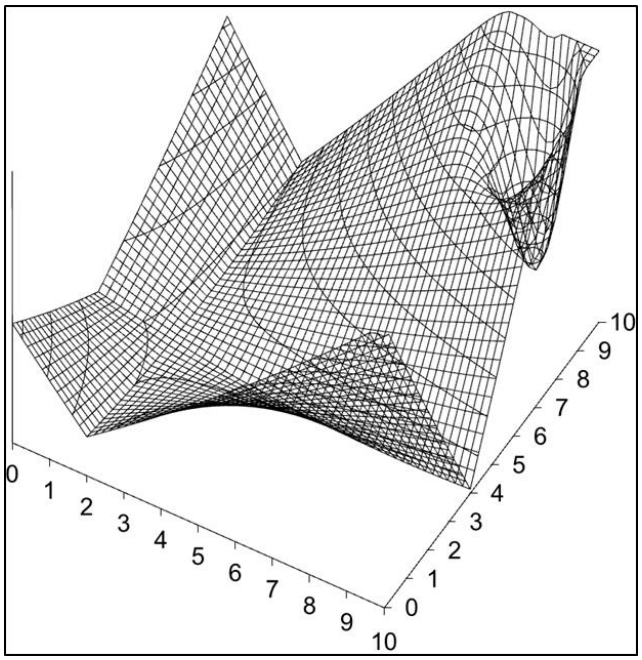


Figure 19: Sharp troughs (F7)

Figure 20 is a Jupitor's eye function (F8) with a twisted slot bottom to the minimum. While it is relatively easy for an optimizer to find the hole, the difficulty is in finding the bottom where the optimum is.

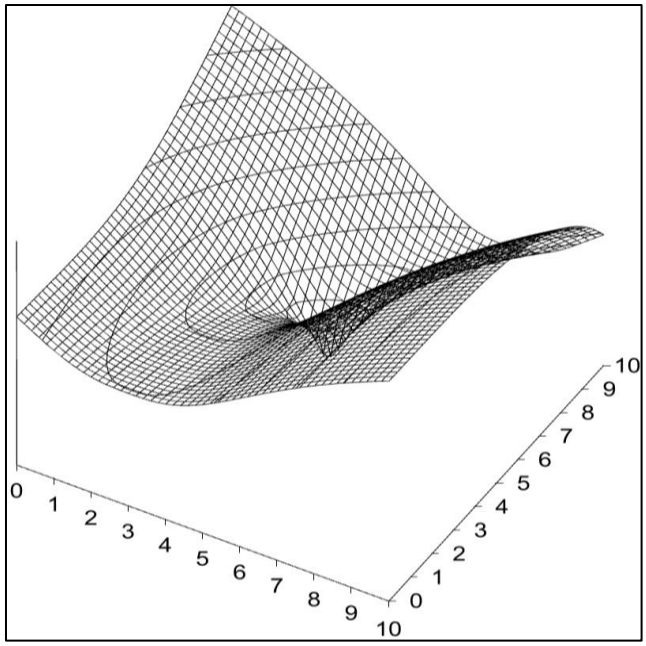


Figure 20: Jupitor's eye (F8)

Details about the test functions F1 through F8 are available in existing literature [2, 43]. Function F9, is a model for separating a mixture of methanol and water. Function F9 has 10 DVs – 9 liquid molar flows and 1 composition. The DVs are interacting, and have hard constraints (compositions strictly between 0 and 1). The OF surface is nonlinear and the global is confined to a narrow DV range.

LF is tested on all the 9 functions listed above, using α values of 0.25, 0.5, 1 and 1.5. The α value indicates the ratio between the reflected window and the original window size formed between the best and the worst player $d(i)_{wb}$. The window sizes are selected by striking a balance between a large α which may lead to instability of the optimizer and a small α which may lead to premature convergence.

The players are initialized in two manners – throughout the DV range, and within a narrow DV range. Often, when there is no prior knowledge about the range of the best DV values, a broad initialization range is used to capture all possible solutions (global initialization). When players

are initialized throughout the DV range, one of the players will be in the vicinity of the optimum (or the global in case of multiple optima), thereby increasing the frequency with which an optimizer can find the optimum. However, there are problems where a solution is expected within a specific DV range but the optimum lies outside the initialized DV range. Therefore, to mimic this situation, players are initialized within a local DV range (local initialization). For the initialization to create a scenario of bounding the optimum or its vicinity during the initialization stage, players are initialized over a DV range of 0 to 10 for both DVs. For all the 2-DV functions the players are initialized locally over a DV range 0 to 1 for both DVs. For F9, in case of global initialization, the players are initialized between a composition range of 0 to 1 and the molar liquid flows are initialized between at 0 to 3*(Feed + Reflux rate). For F9 in the case of local initialization, the players are initialized between a composition range of 0.3 to 0.4 and the molar liquid flows between at 1.1*(Feed + Reflux rate) – 1.2*(Feed + Reflux rate). The optimizer is initialized 1000 times from random starts with each α value so that the results represent broad expectations. A traditional DV-based convergence criteria is employed in all the cases where the optimizer stops if $\Delta DV <= 0.00001$.

4.2 Steady state binary distillation modeling

The purpose of the steady state binary distillation model is for demonstration of LF as a feasible method. The purpose is not to develop a new solution method for solving distillation models. Therefore the model is used to demonstrate proof-of-concept of LF and the accelerated convergence modification. Section 4.1 has already detailed the procedure used to test LF and the accelerated convergence modification on the steady state binary distillation models.

4.3 Dynamic binary distillation modeling

The dynamic binary distillation model is initialized and allowed to reach steady state. Subsequently, the dynamic model is tested for open loop responses by carrying out the following changes:

1. Step change of reflux flow rate by +10% of range and -10% of range
2. Step change of reboiler heating power by +10% of range and -10% of range
3. Step change in feed flow rate by +10% of range and -10% of range
4. Step change in feed composition by +10% of range and -10% of range

When each of the step changes above is made, the other inputs to the simulation model are maintained at a constant.

4.4 Nonlinear model predictive control of distillation

Standard controller tests such as setpoint tracking (servo mode), disturbance rejection (regulatory mode), and constraint handling were performed. These tests establish the credibility of the controller and demonstrate the LF ability to handle NMPC. Table 9 details the operating ranges, and constraints used in the NMPC simulation. The operating ranges were determined based on the existing limits on the Unit Operations Lab. There is a lower bound for the reflux rate of 0.001 lbmol/min, so that the column does not encounter a no-reflux condition. There is a lower bound for the reboiler duty of 160 Btu/min. In the UOL distillation column, below a reboiler duty of 160 Btu/min the vapor rates are very low and lead to a loss of hydrodynamic seal on the trays and severe weeping is observed. Rate of change constraints on the MVs were also imposed to mimic industrial operation, where large changes in the MVs such as steam rate to the reboiler could cause a drop in the steam header pressure, affecting other unit operations that withdraw steam.

Table 9: NMPC operating ranges and constraints

Reflux rate	0.001 to 0.015 lbmol/min
Reboiler duty	160 to 300 Btu/min
Feed rates	0.005 to 0.015 lbmol/min
Rate of change constraint on MVs	Reboiler duty <= 5 Btu/min Reflux rate <= 0.001 lbmol/min

In order to mimic process reality, Box-Muller noise [44] is added to the process simulation. Box-Muller noise is normally and independently distributed as in Equation (38).

$$rn = \sigma\sqrt{-2\ln(r_1)}\,\sin(2\pi r_2) \tag{38}$$

The σ in Equation (38) is the standard deviation of the desired noise. Based on data collected from the UOL distillation column, the range of the noise in measurement of temperature was ± 0.5°C. From the T-x-y diagram of a methanol water system, a ± 0.5°C deviation in temperature translated to approximately ± 0.001 to ± 0.005 mole fraction of methanol depending on the temperature range. The lower bound of ± 0.001 mole fraction of methanol was used as the range for calculating σ in Equation (38). σ is approximated as a fifth of the range. Using experimental data to determine variability ensures that the noise added by the Box-Muller method is within the limits of variability that a measurement sensor would create. However for the purpose of illustration and clarity, some tests do not incorporate noise in the process simulation. Subsequent sections describe the tests.

4.4.1 Bumpless transfer

When a controller is transferred from open loop or Manual mode to closed-loop or Auto mode, the transfer should be bumpless to avoid process upsets. Initially, the controller is placed in Manual mode and allowed to reach steady state. The controller is then placed in Auto mode to test if the setpoint is retained by the controller and there are no upsets in the CV or MV.

4.4.2 Setpoint tracking

Setpoint changes in both the CVs (top and bottom tray composition) are made to test the controller. LF uses both MVs (reflux rate and reboiler duty) to move the process towards the desired state.

4.4.3 Controller aggressiveness

The tuning parameters are adjusted to demonstrate controller aggressiveness. Additionally, the equal concern factors are also adjusted such that deviations in one of the CVs are weighted with a lower equal concern (meaning lesser tolerance on deviations from setpoint).

4.4.4 Disturbance rejection

The disturbance rejection capability of the controller is tested in two ways. One, by creating a disturbance in the feed flow rate, and second, by creating a disturbance in the feed composition. When the feed flow rate or feed composition is changed, the controller is expected to adjust the MVs to keep the CVs at their setpoints.

4.4.5 Constraint handling

Under closed-loop conditions, when setpoint changes were made, a physically unrealizable setpoint was reached. At the physically unrealizable setpoint, one or both of the MVs reach their

operating boundaries and the controller hits a constraint. After retaining the unrealizable setpoint for a duration that is nearly a settling time, a realizable setpoint change was made to test if the controller has windup or, if it responds immediately by moving out of the constrained region.

CHAPTER V

RESULTS AND DISCUSSIONS

5.1 Accelerated Convergence Improvement

5.1.1 Global Initialization

Table 10 summarizes the average number of leapovers to converge at the optimum obtained by initializing the players randomly over the entire feasible DV space, with various leap-to window size factors. For functions with multiple optima – F1, F2, F3, and F7 the average number of leapovers presented in Table 10 represents the number of leapovers to global optimum after one of the players has landed in the vicinity of the global. The vicinity of the global is identified when, one player leaps to an OF value which is lesser than the 2nd best OF. For functions with a single optimum – F4, F5, F6, F8 and F9, the DV initialization represents that one of the players is always at the vicinity of the optimum, so the values represent the average leapovers to optimum after initialization. The results obtained using an α value of 1.0 represent the base case of LF. For α values lower than 1.0, the general trend is that the average number of leapovers to convergence is significantly less than the base case of LF. This means that with α values of 0.5 and 0.25, LF takes fewer leapovers, i.e. lower computational burden, and accelerated convergence. On the other hand, for α values greater than 1.0, LF takes more number of leapovers than the base case, indicating that it takes greater computational burden to find a solution. Figure 21 confirms that

the number of leapovers to convergence is linearly dependent on the reciprocal of ln(0.5* α) as was anticipated from Equation (10). Additionally, Figure 21 also indicates that the slope of the trend lines corresponding to each function is less than the anticipated slope of (M-1)*ln(ε).

Table 10: Average number of leapovers to convergence (after locating the vicinity of the global) for players initialized by encompassing the global

α	**0.25**	**0.5**	**1**	**1.5**
-1/ln(0.5*α)	0.4809	0.7214	1.4427	3.4761
F1	92	112	185	330
F2	82	103	177	313
F3	85	111	182	327
F4	91	120	197	358
F5	126	121	207	365
F6	91	113	180	317
F7	164	202	326	582
F8	147	183	302	543
F9	14870	15861	28333	39618

As mentioned in Section 3.1, lower number of leapovers does not guarantee that the optimum solution was found at the same or higher probability as the base case with α value of 1.0. Small α values accelerate convergence, while large α values improve exploration but run the risk of leading to instability. By plotting the PNOFE of all the functions against -1/ln(0.5*α) we analyze the impact of α on the combined factors for computational efficiency and robustness. Figure 22 is a plot of the PNOFE obtained for the test cases where the players are initialized by encompassing the global. Based on Figure 22, for functions F1-F9, for six out of the nine test cases, PNOFE is

the least when α value is 0.5. On either side of α = 0.5, PNOFE is higher. However, for F9, which has 10 DVs, interacting and hard constraints, it appears that α = 1.0 is the best balance of speed and robustness. It is pertinent to mention that the CDF (which is indicative of the frequency with which LF found the true solution) for α = 0.5 is 94.6% while for α = 1.0 is 99.9%. For most practical purposes, 95% is an acceptable frequency of success for an optimizer. On the same note, the ANOFE for α = 0.5 was only about 2.5% lesser than the ANOFE with α = 1.0. Therefore, while the PNOFE with α = 1.0 was lower than the PNOFE with α = 0.5, the performance of the optimizer, for all practical purposes, can be considered equivalent.

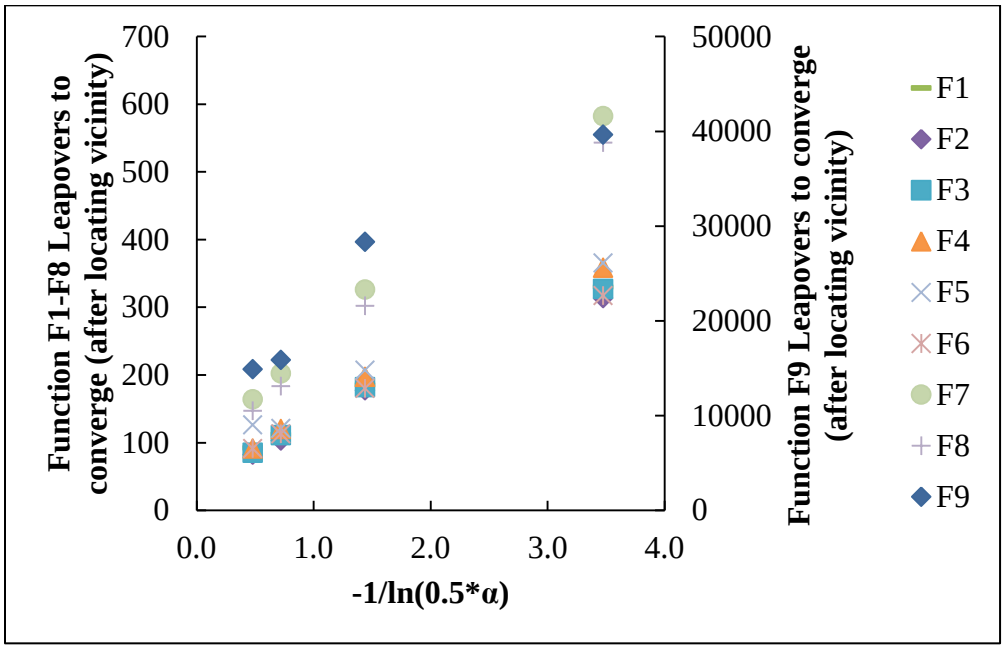


Figure 21: Linear relation between α and average number of leapovers to convergence after locating vicinity of global

For functions F7 and F8, the reduction in ANOFE is significant, but does not translate into a lower PNOFE because of a decrease in the CDF. While Figure 21 and Table 10 supported the general hypothesis that when α is smaller convergence will be faster, however, Figure 22 suggests that there is a lower threshold value for α. Beyond the lower threshold value, convergence was achieved faster, but CDF values were significantly lower than when α was 0.50. Further

investigation revealed that this was because of premature convergence of LF when α<0.5; the algorithm proceeds towards the optimum, but converged prematurely. This was because, α was too small to allow significant exploration that would have directed the search towards the optimum. Therefore, when the players are initialized globally, using an α of 0.5 best balances robustness and efficiency.

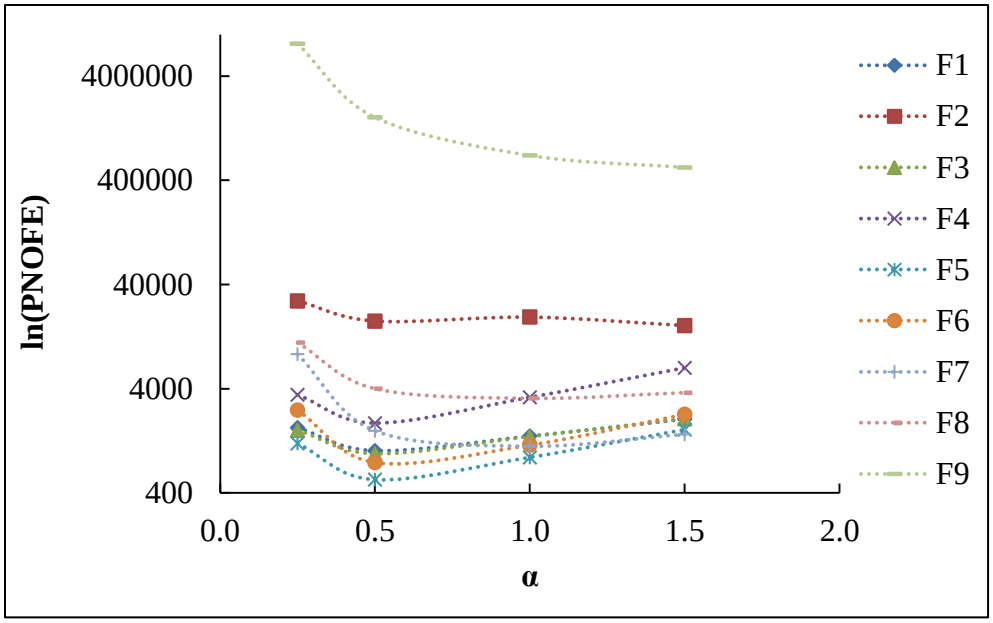


Figure 22: PNOFE comparison for initialization encompassing global for α = 0.25, 0.50, 1.00, 1.50

5.1.2 Local initialization

Table 11 summarizes the average number of leapovers to stop at the optimum obtained by initializing the players at a local DV range of the feasible DV space, with different leap-to window size factors.

For functions F1 and F2, the global optimum is away from where the players are initialized locally. For function F1, LF failed to find the global with α = 0.25, 0.5 and 1, and with function F2, LF failed to find the global with α = 0.25 and 0.5. F1 and F2 [4, 10] have a flat surface around the area of initialization, thereby making it difficult for LF to move away from the flat region to a

region with curvature. This essentially means that when the players are initialized on a flat surface, $\alpha < 1$ traps the players. However, when the players are initialized locally on a surface with curvature, for instance F3 and F7, LF successfully moves towards the solution and converges at the optimum.

Table 11: Average number of leapovers to stopping (after locating the vicinity of the global) for players initialized at a local DV range

α	**0.25**	**0.5**	**1**	**1.5**
-1/ln(0.5*α)	0.4809	0.7214	1.4427	3.4761
F1	-	-	-	335
F2	-	-	183	288
F3	89	115	190	339
F4	141	148	222	397
F5	268	228	310	453
F6	187	134	192	321
F7	271	226	345	615
F8	261	227	335	584
F9	17800	16886	29429	39202

Figure 23 is a plot of the number of leapovers to convergence after locating the vicinity of the global vs the reciprocal of $-\ln(\alpha/2)$. For functions F5, F7, F8 and F9 the number of leapovers to stopping is higher with $\alpha = 0.25$ than with 0.5. This is because, when all the players are initialized locally, they first need to move towards the global, then a player leaps to the vicinity of the global and draws the other players towards the global. This involves more leapovers, especially when

the surface has flat regions and irregularities, which then force the players to spend computational effort moving away from the aberrations such as cliffs. The players are also forced to spend additional computational effort because of the local initialization range itself, when they are far away from the global. However, in general if $\alpha > 0.25$ the relation between α and the number of leapovers continues to be linear. Additionally, Figure 24 also indicates that the slope of the trend lines corresponding to each function is less than the anticipated slope of $(M-1)*\ln(\varepsilon)$. Figure 24 is a plot of the PNOFE for the different functions with local initializations. For functions (F3, F4, and F6) $\alpha = 0.5$ has the lowest PNOFE. However, $\alpha = 1.0$ (the base case) has the lowest PNOFE for some functions (F5, F7, F8, and F9). This is because, for function F7, F8, and F9 when $\alpha = 0.50$, the CDF is lower than the CDF achieved with $\alpha = 1.0$. Therefore, while the reduction in the number of leapovers is significant for $\alpha = 0.50$ compared to $\alpha = 1.0$, this does not translate into a reduced PNOFE. Therefore, to generalize, when the players are initialized locally, $\alpha = 1.0$ may be chosen as the best balance between robustness and efficiency.

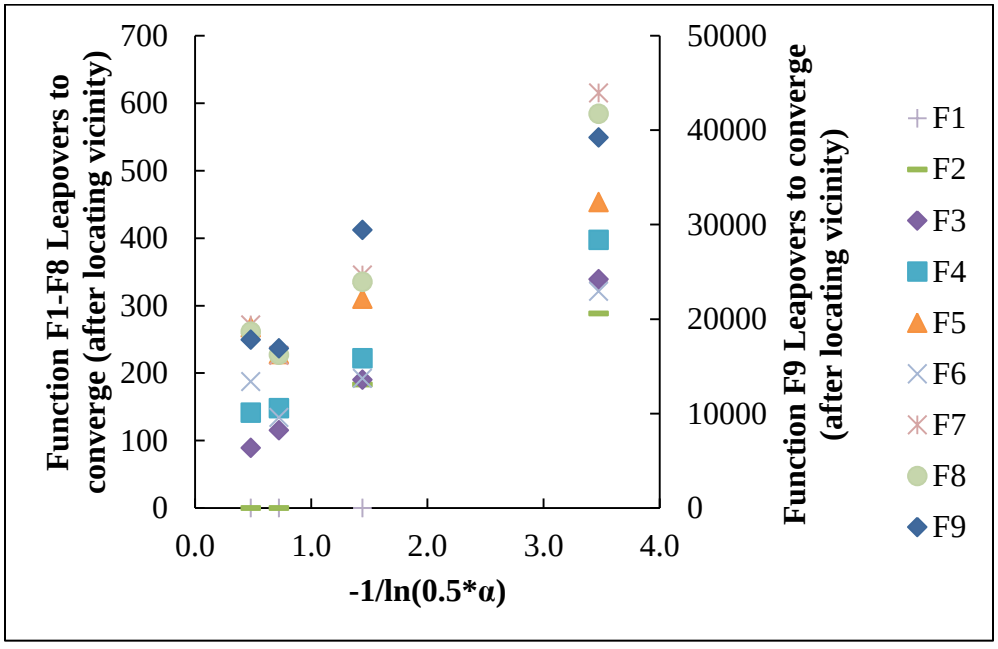


Figure 23: Leapovers vs -1/ln(0.5*α) for local initialization

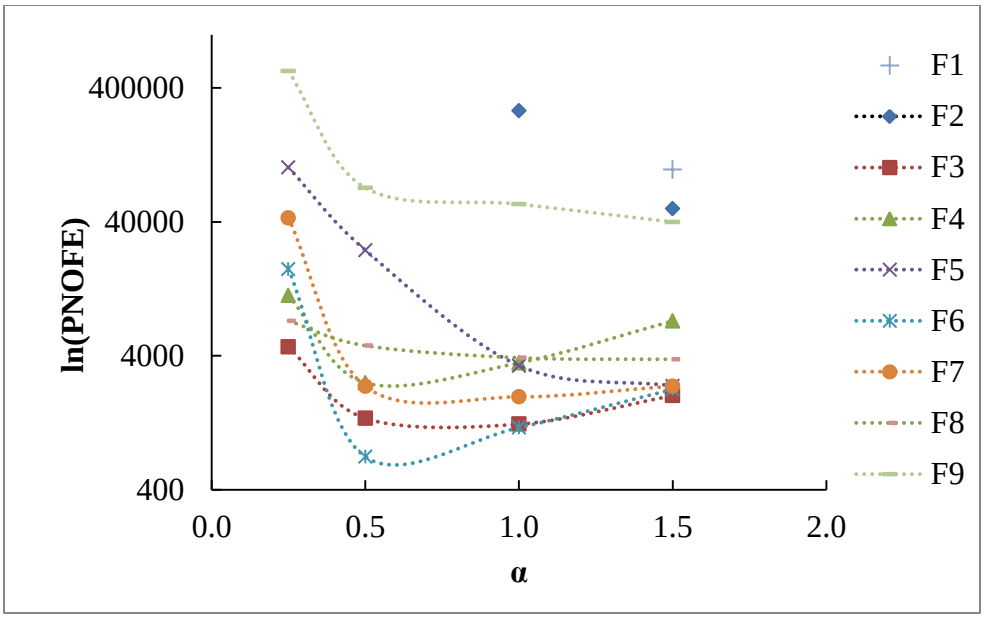


Figure 24: PNOFE comparison for local initialization

5.2 Steady state binary distillation modeling

The steady state binary distillation model is to demonstrate proof-of-concept of nonlinear process modeling, and to extend the steady state model to a dynamic model. One random initialization of the steady state model is detailed below to elucidate the functioning of the model. The model input parameters used to simulate the steady state model are shown in Table 12.

A five stage column was simulated and explained below. For the purpose of computational nomenclature, the condenser is considered a stage (not an equilibrium stage). The reboiler is considered as the last stage for simulation (and is an equilibrium stage). There are five equilibrium stages within the column and the reboiler provides an additional stage of separation. The DVs are the top tray liquid mole composition and the liquid mole flows of all the five trays. For a seven stage simulation there are six DVs. The initialization range for the DVs and the optimizer parameters used are specified in Table 13.

Table 12: Model inputs used to simulate steady state model results

Model Parameter	**Units**
Number of stages	7 (1 condenser, 5 stages in column, 1 reboiler)
Feed stage	3
Feed mole flow rate	0.008 lbmol/min
Feed mole fraction (of methanol)	0.50
Reflux mole flow rate	0.006 lbmol/min
Reboiler duty	200 Btu/min
Feed temperature	30 °C
Reflux mole fraction (of methanol)	0.85
Reboiler mole fraction (of methanol)	0.05

Table 13: DV initialization range and LF parameters

DV	**Initialization Range**
Liquid mole fraction	0-1
Liquid mole flows	Reflux – 2*Reflux+Feed
LF constants	**Value**
α	0.50
Maximum iterations	25,000
Stopping criteria	$\Delta(\text{Best-Worst})DV_x \leq 10^{-10}$; $\Delta(\text{Best-Worst})DV_L \leq 10^{-10}$

Figure 25 shows the progression of the liquid mole composition DV from a random initialization. Starting from a random composition, after 1000 leapovers, LF reaches the composition that nearly closes the material and energy balance. However, the stopping criterion of $\Delta DVs \leq 10^{-10}$ is

not met until 2000 leapovers when LF eventually stops. The stopping criteria is "tight" because the dynamic model uses the steady state model as a precursor and requires repeatable values that are not confounded by a "coarse" stopping criteria.

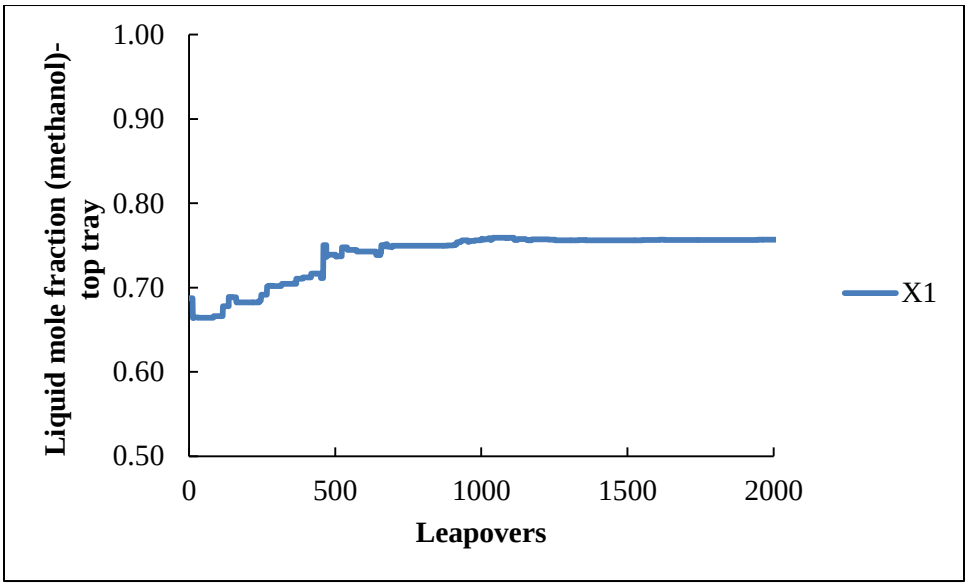


Figure 25: Progression of liquid mole fraction (DV) with leapovers for one random initialization

Figure 26 shows the progression of the liquid mole fraction DVs from random initialization. Again, after 1000 leapovers, LF is near the mole flows which closes the material and energy balances. Beyond 1000 leapovers, LF is working towards meeting the "tight" stopping criteria.

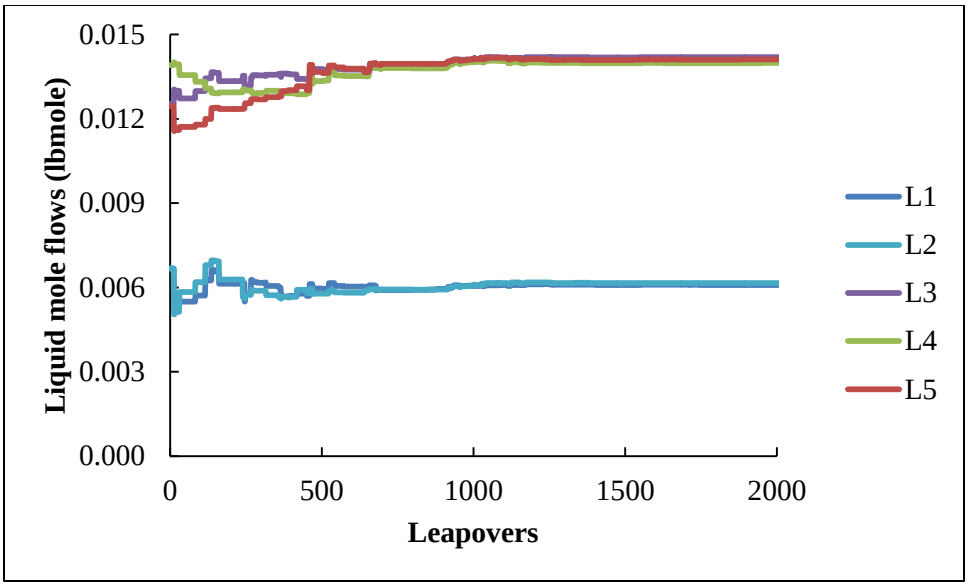


Figure 26: Progression of liquid mole flows (DV) with leapovers for one random initialization

Figure 27 shows the progression of the OF for the best and worst player over the leapovers. OF-best is the player with the lowest OF value and OF-worst is the player with the highest OF value. Range is the difference between OF-best and OF-worst. After 1000 leapovers, OF-best and OF-worst reach nearly the same values. Therefore, Figure 27 corroborates the observations from Figure 25 and Figure 26 where the DV values reached close to the solution values at about 1000 leapovers. Based on Figure 27 it is also evident that the OF values of the worst player at stopping are at least three orders of magnitude lower than the initialization. The Range values at stopping, however, are at least ten orders of magnitude lower than initialization. This demonstrates the ability of LF to handle nonlinear process models where the initialization is several orders of magnitude higher than the final solution.

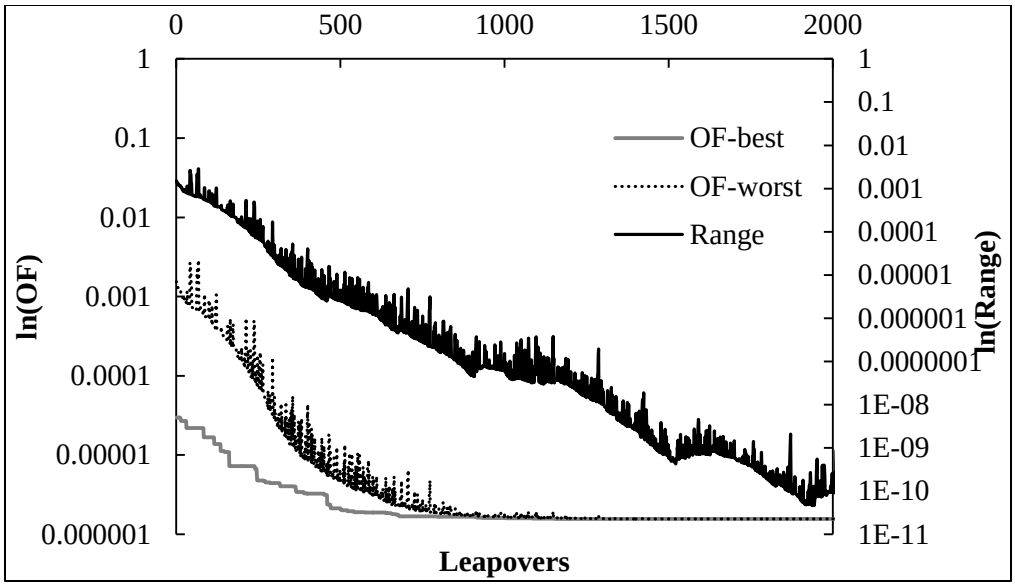


Figure 27: Progression of best and worst players and Range with leapovers

5.3 Dynamic binary distillation modeling

The dynamic model for a 5 stage distillation column with a reboiler and total condenser is initialized and allowed to reach steady state values before testing for open loop responses. For the purpose of simulation, the reboiler has a reduced liquid volume of 0.051 ft^3 instead of 0.51 ft^3. This reduced volume allows the reboiler to respond faster to changes, and reach steady state faster than with the original reboiler volume of 0.51 ft^3. The initial steady state values are shown in Table 14.

Figure 28 -Figure 35 show the dynamic response of the process over a nominal operating range. Figure 28 and Figure 29 show the dynamic response of the model to step changes in reflux rates. Notice that the reboiler and accumulator vapor compositions lag the bottom and top tray compositions respectively. While the top tray vapor and accumulator liquid compositions reach the same values, the bottom liquid and the reboiler liquid compositions are not the same because the reboiler is an equilibrium stage of separation. The gains for the top ($K_{top\text{-}reflux}$) and bottom ($K_{bot\text{-}reflux}$) compositions with respect to a change in the reflux flow rate (+10% and -10% of

Table 14: Steady state initialization of dynamic model

Model Parameter	**Units**
Number of stages	7 (1 condenser, 5 stages in column, 1 reboiler)
Feed stage	3
Feed mole flow rate	0.008 lbmol/min
Feed mole fraction (of methanol)	0.50
Reflux mole flow rate	0.006 lbmol/min
Reboiler duty	200 Btu/min
Feed temperature	30 °C
Steady state reflux composition (mole fraction of methanol)	0.8400
Steady state bottom tray composition (mole fraction of methanol)	0.0211
Steady state reboiler composition (mole fraction of methanol)	0.0045

range) are not the same. $K_{top\text{-}reflux}$ is 39.8 mole fraction/lbmol.min^{-1} and $K_{bot\text{-}reflux}$ is 22.0 mole fraction/lbmol.min^{-1} for a +10% change in reflux rate. However, $K_{top\text{-}reflux}$ is 61.8 mole fraction/Btu.min^{-1} and $K_{bot\text{-}reflux}$ is 6.1 mole fraction/Btu.min^{-1} for a -10% change in reflux rate. Around a nominal operating range, a step change in reflux rates produces a gain change that is very different for a +10% change in reflux rate and a -10% change in reflux rate providing a sense for the inherent nonlinearity of the process. Figures 30 and 31 show the dynamic response to step changes in reboiler duty; Figures 32 and 33 show the dynamic response to step changes in

measured disturbance (feed rate); Figures 34 and 35 show the dynamic response to step change in measured disturbance (feed composition).

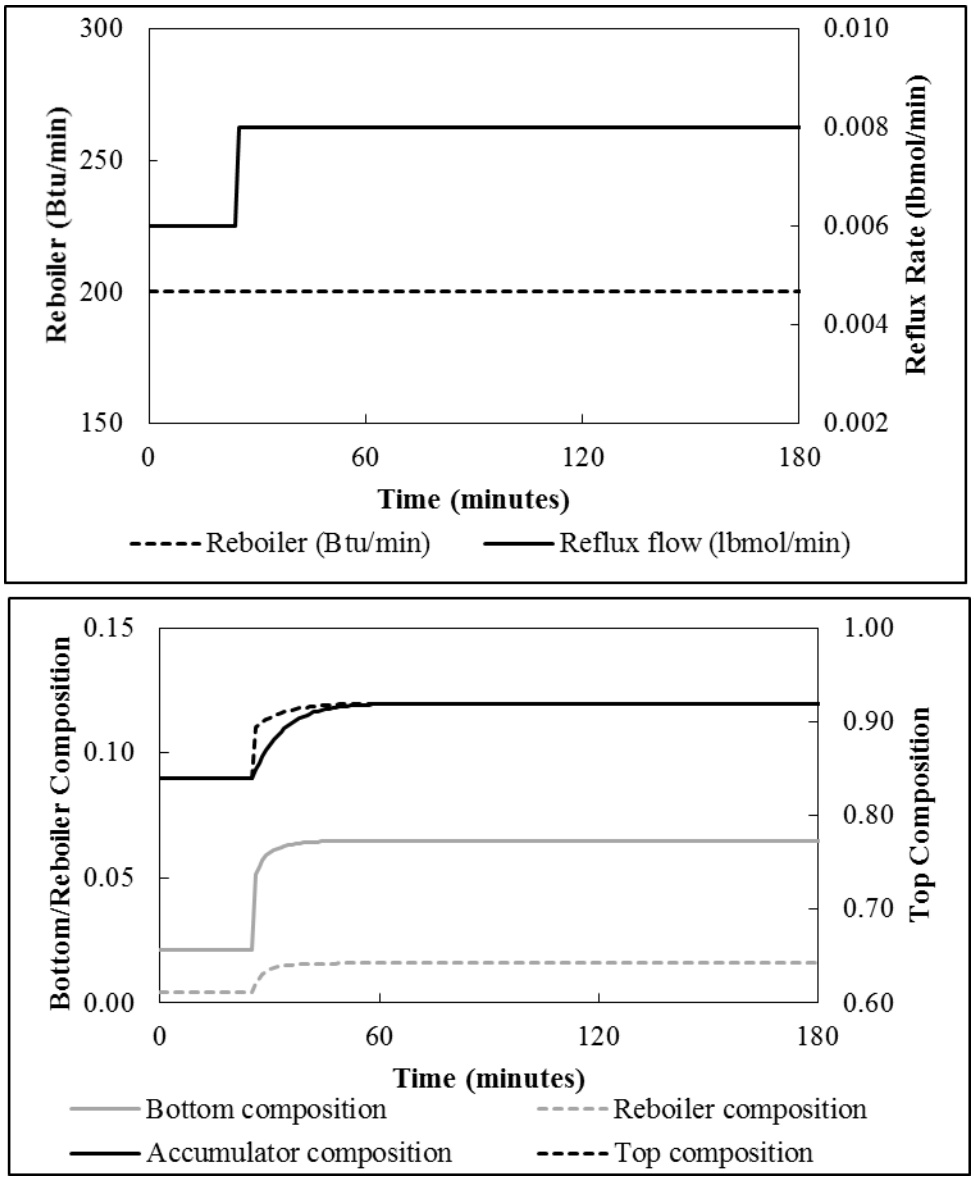


Figure 28: Dynamic response to step change in reflux rate (+10% of range)

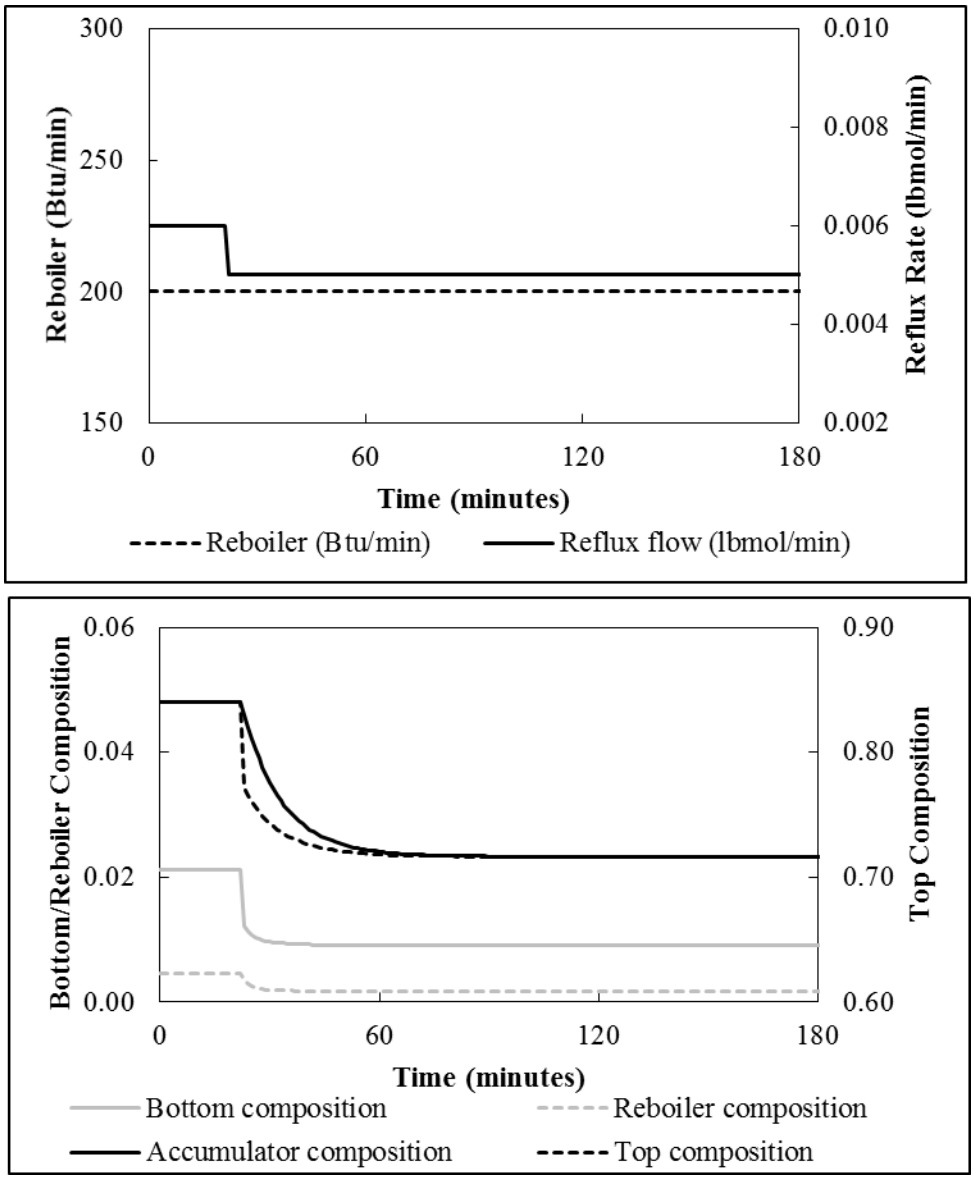


Figure 29: Dynamic response to step change in reflux rate (-10% of range)

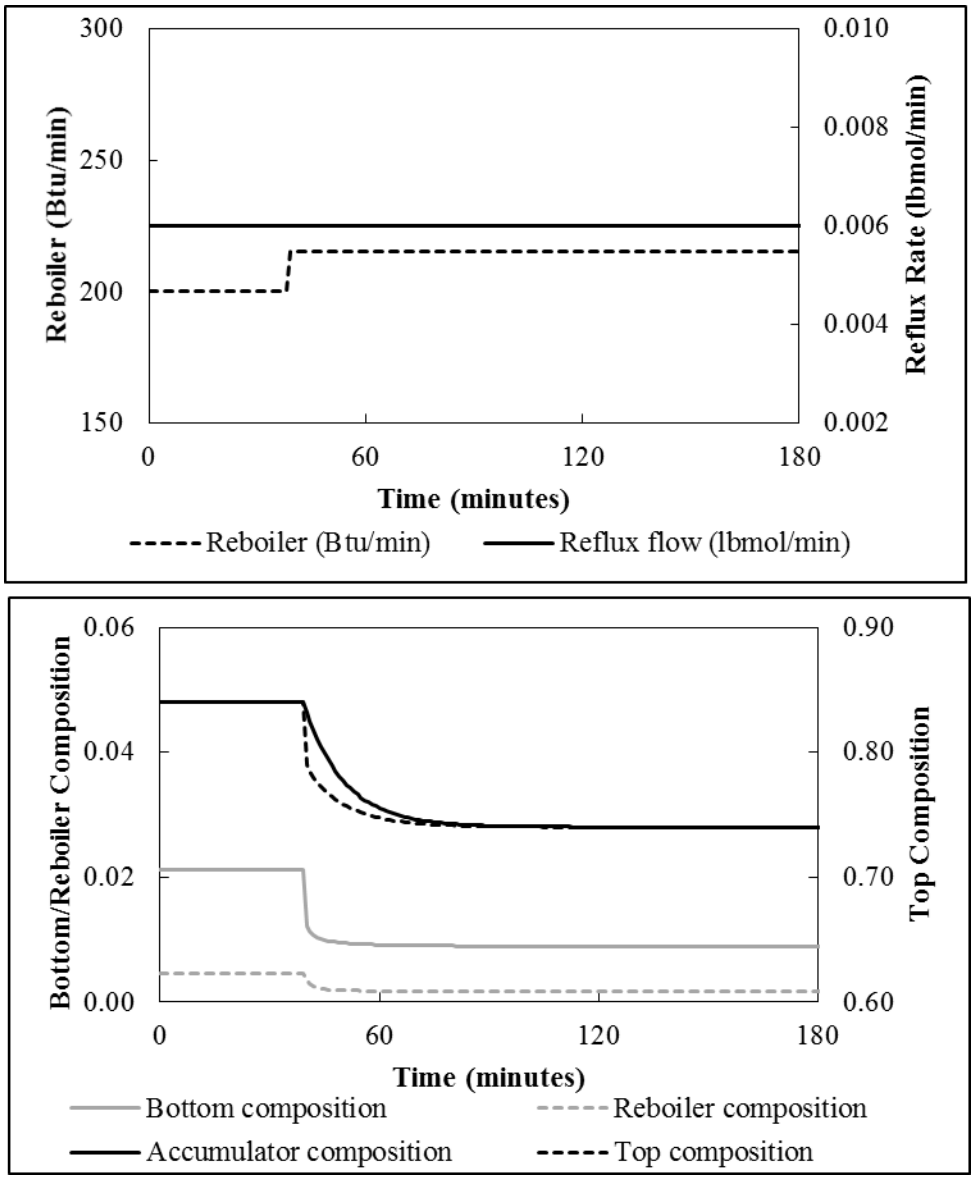


Figure 30: Dynamic response to step change in reboiler duty (+10 % of range)

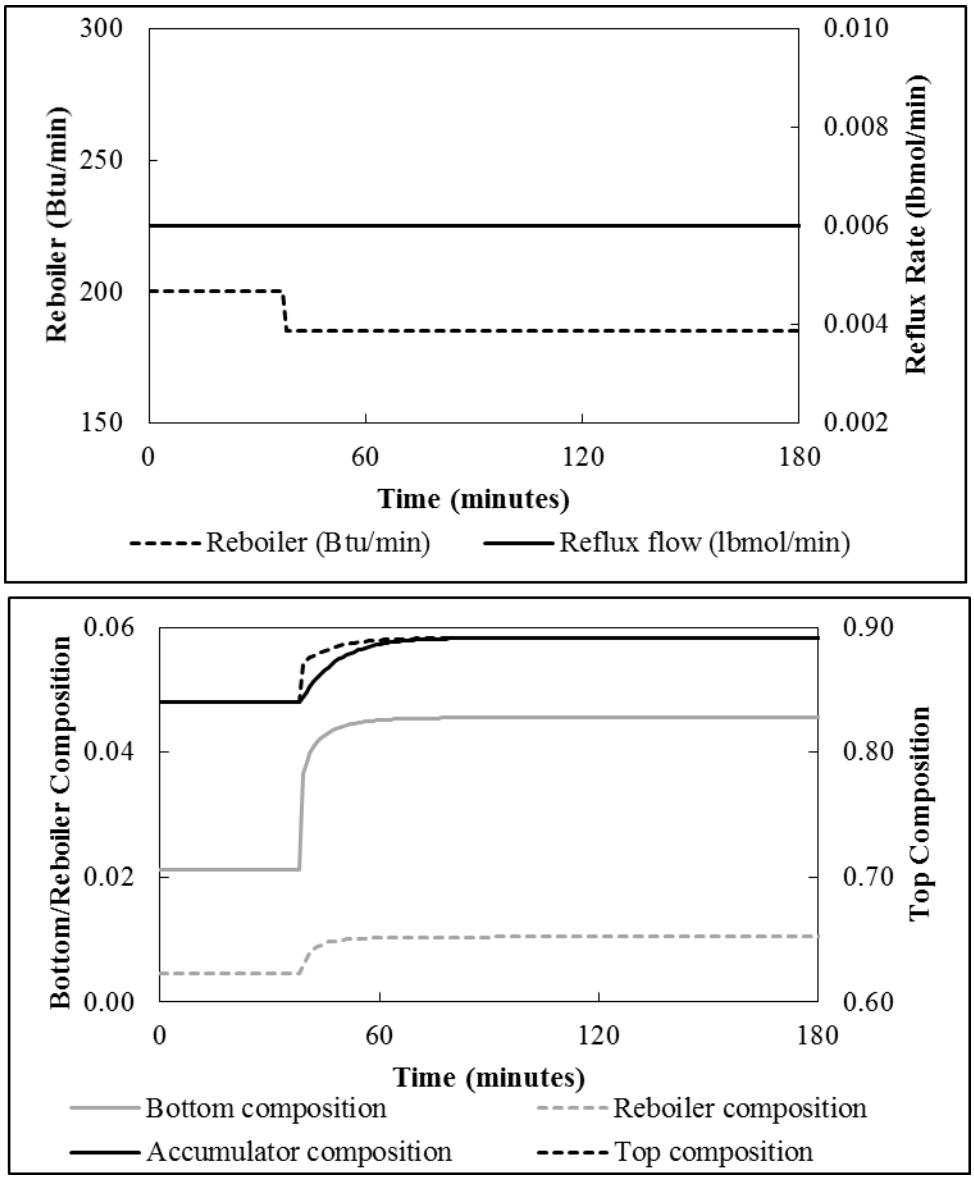


Figure 31: Dynamic response to step change in reboiler duty (-10 % of range)

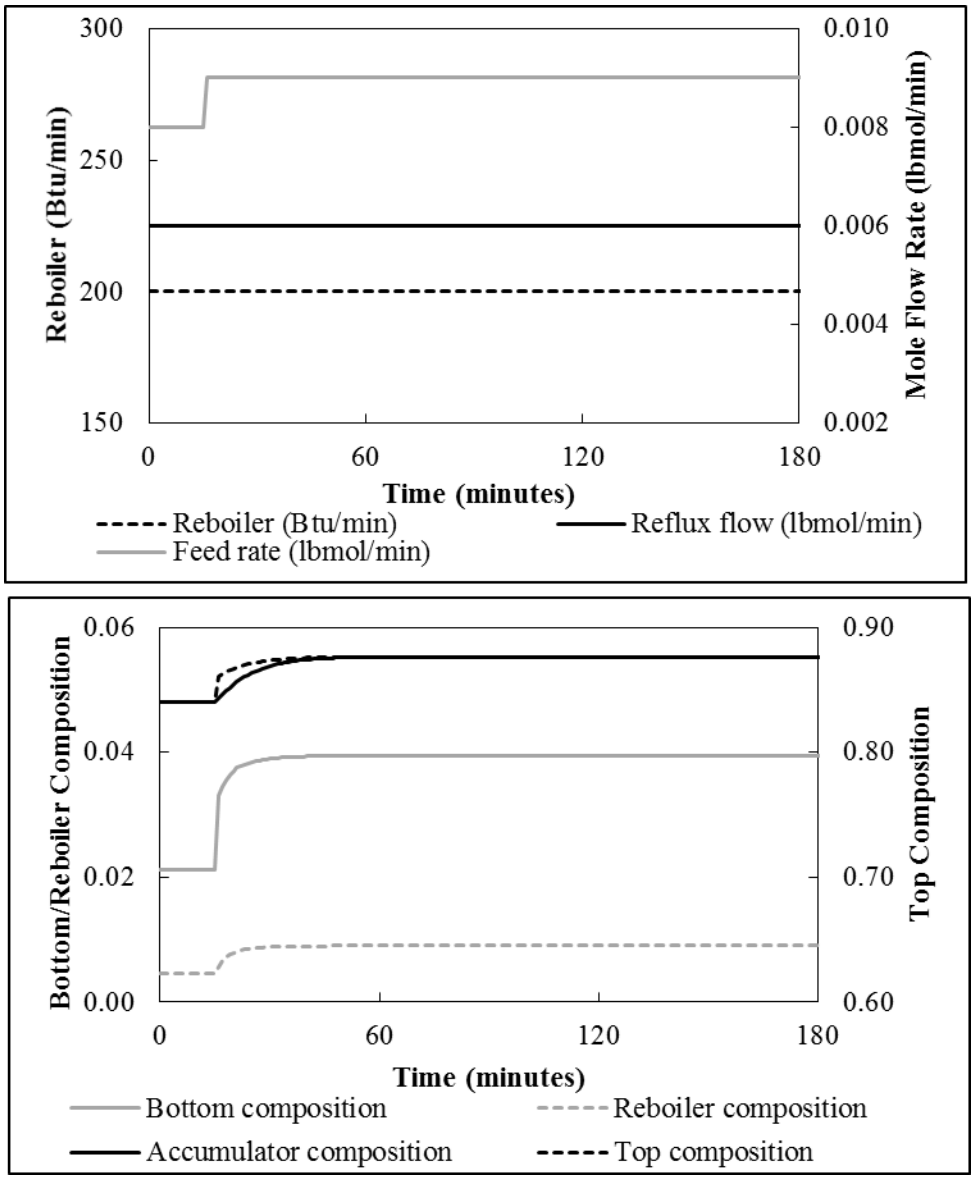


Figure 32: Dynamic response to step change in measured disturbance - feed rate (+10 % of range)

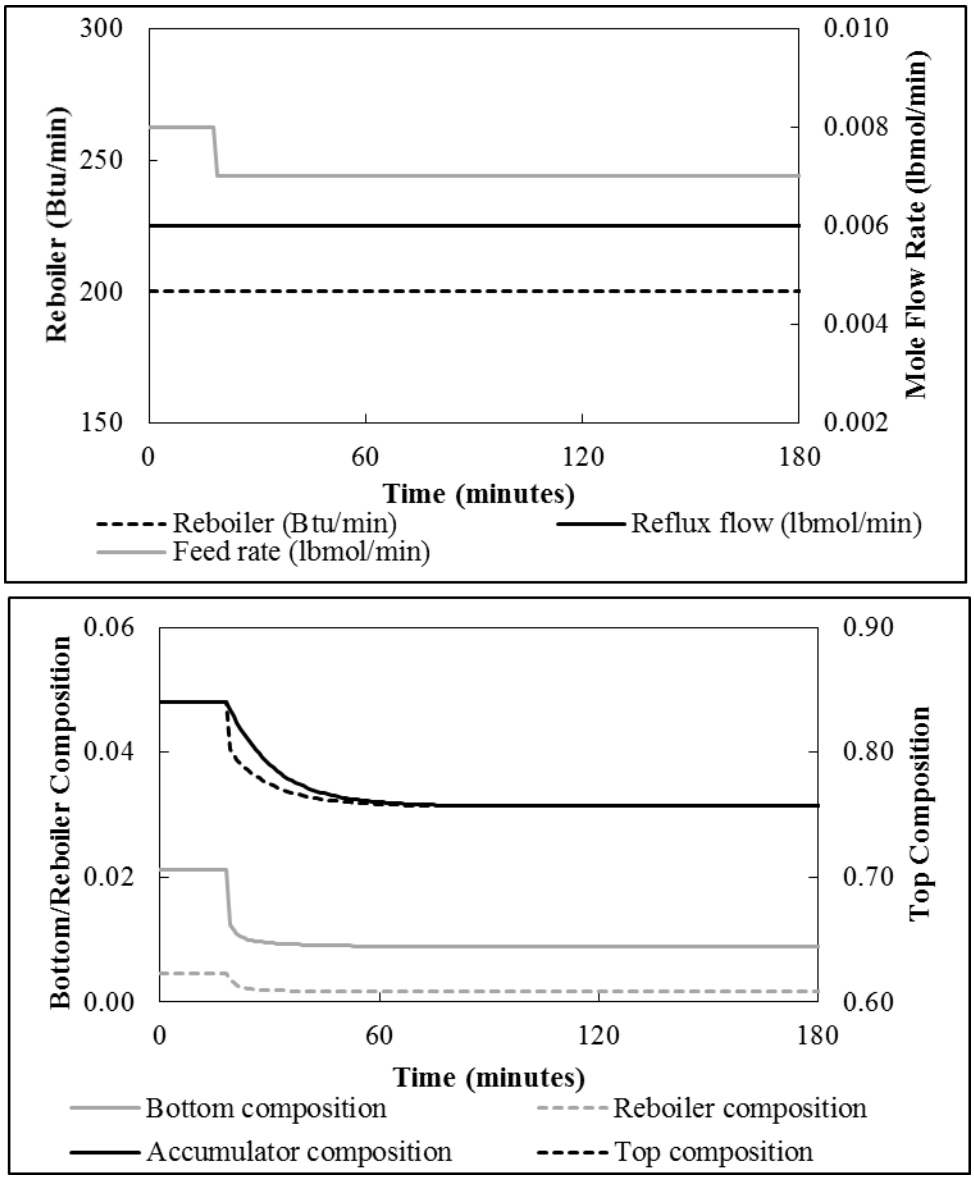


Figure 33: Dynamic response to step change in measured disturbance - feed rate (-10 % of range)

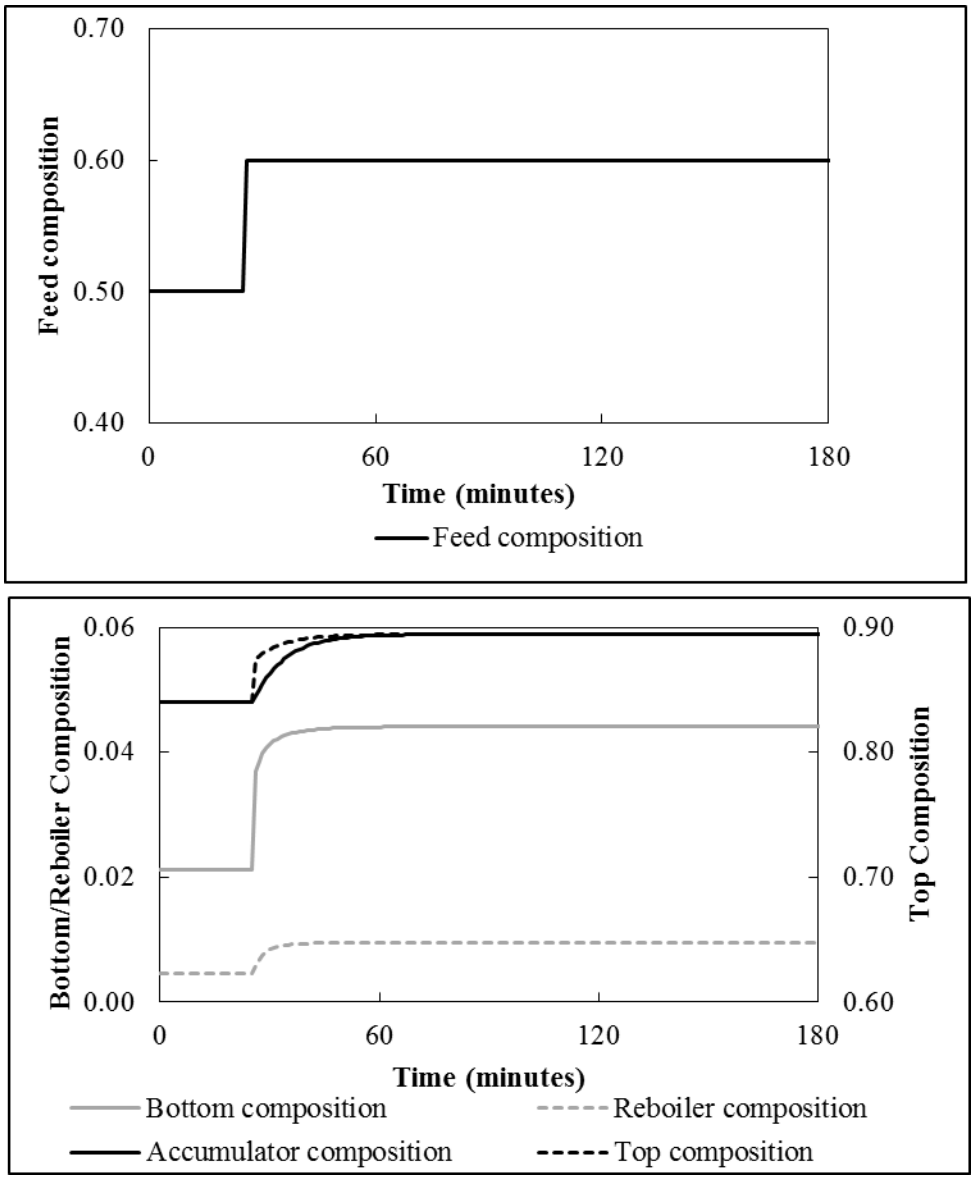


Figure 34: Dynamic response to step change in measured disturbance - feed composition (+10 % of range)

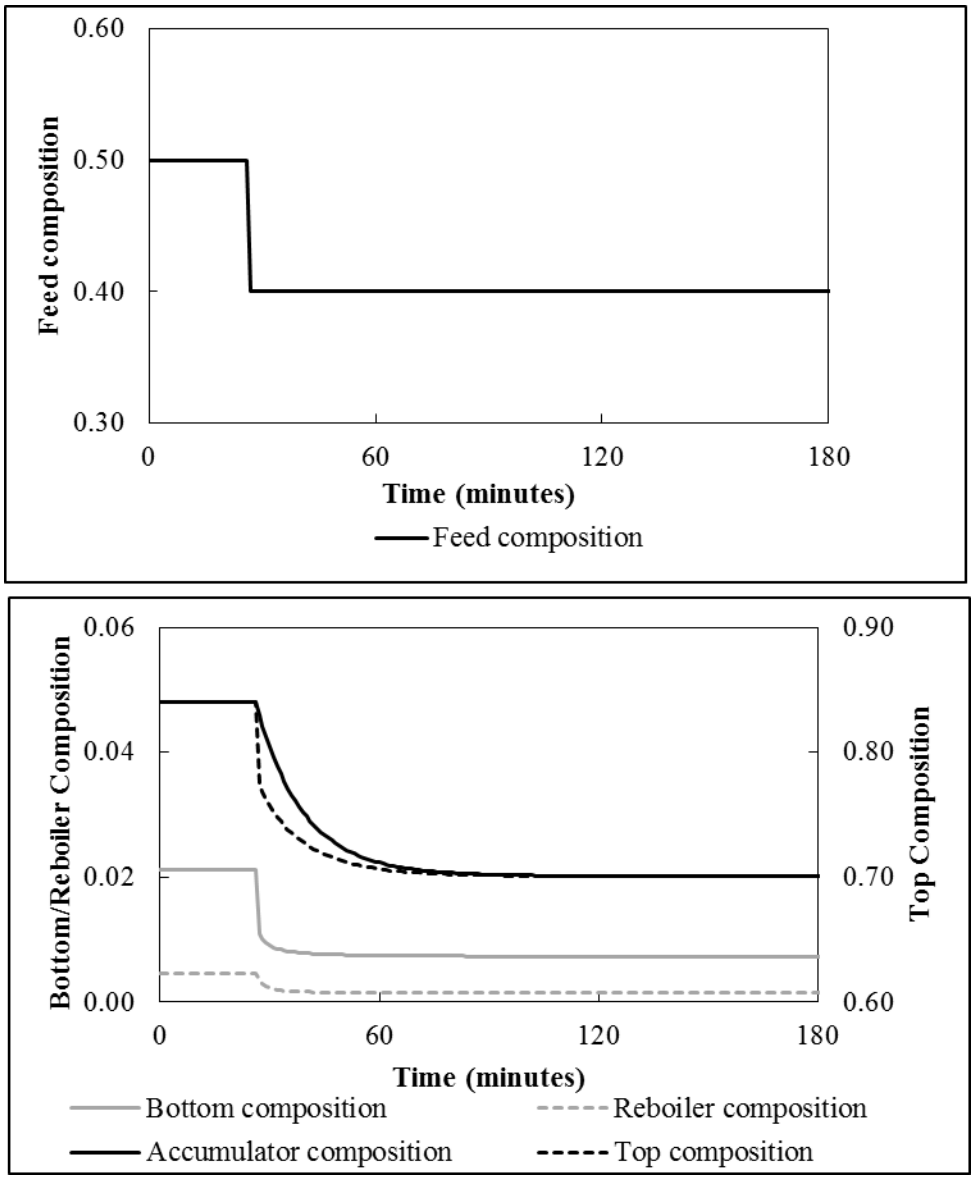


Figure 35: Dynamic response to step change in measured disturbance - feed composition (-10% of range)

Based on Figures 28-31 the steady state gains of the top and bottom tray compositions with respect to the reboiler duties and reflux rates are listed in Table 15. Within a nominal operating range, the gains can double or triple indicating how significant the nonlinearity is even within a limited operating range. The purpose of the preceding analysis was to demonstrate nonlinearity and not to determine the steady state gains for NMPC. One of the advantages of using first principles models for NMPC is it does not require step testing to determine steady state gains.

Table 15: Steady state gains based on open loop analysis

Variable	$\mathbf{K_{top}}$	$\mathbf{K_{bottom}}$
Reflux rate (+10%)	39.8 mole fraction/lbmol.min^{-1}	61.8 mole fraction/lbmol.min^{-1}
Reflux rate (-10%)	22.0 mole fraction/lbmol.min^{-1}	6.1 mole fraction/lbmol.min^{-1}
Reboiler duty (+10%)	6.6 x 10^{-3} mole fraction/Btu.min^{-1}	3.4 x 10^{-3} mole fraction/Btu.min^{-1}
Reboiler duty (-10%)	8.1 x 10^{-4} mole fraction/Btu.min^{-1}	1.6 x 10^{-3}mole fraction/Btu.min^{-1}

5.4 Nonlinear model predictive control

The tests for demonstrating LF on NMPC simulation of a binary distillation column is carried from an initial closed-loop steady state with LF optimizing for a set of three future control moves.

5.4.1 Bumpless Transfer

Figure 36 demonstrates bumpless transfer. This simulation is demonstrated on a noiseless simulation for the purpose of clarity and illustration. In

Figure 36 the controller was initially placed in manual mode (open-loop steady state). At 30 minutes, the controller is shifted to Auto mode (closed-loop steady state). There are no upsets in the CVs (top tray and bottom tray composition) and the MVs (reboiler duty and reflux rate). The LF optimizer determines MVs that retain the open-loop steady state values after being transferred to closed-loop.

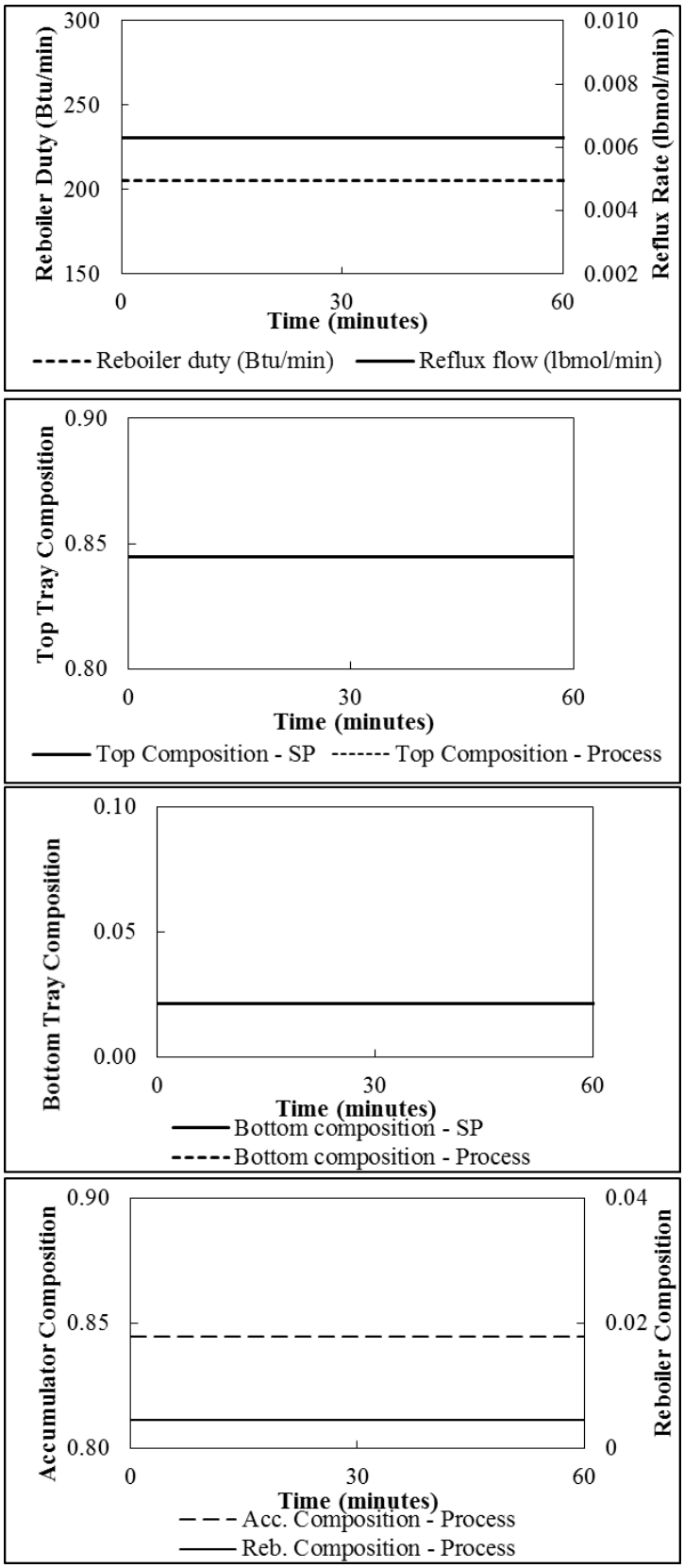


Figure 36: Bumpless transfer

5.4.2 Setpoint tracking and controller aggressiveness

Table 16 lists the cases studied for servo mode (setpoint tracking). Cases 1*, 1 and 2 were conducted starting with a feed rate of 0.008 lbmol/min and feed composition of 0.5 mole fraction of methanol. Case 1* demonstrates controller performance on a noise-less simulation, while Cases 1-6 demonstrate controller performance on a simulation with noise added to the process model.

In Case 1* the setpoint was changed at 97 minutes. The controller results are shown in Figure 37. On the top plot of Figure 37 that shows both the MVs, immediately after the setpoint was changed, the reflux rate is raised to push the process towards the new setpoint before it backs off making several moves and settling down at the new reflux rate of 0.0056 lbmol/min. The reboiler duty drops steadily before settling down at the new reboiler duty of 182.7 Btu/min. The second and third plots of Figure 37 show the CVs, which take about 60 minutes to settle at the new steady state. At each controller sample, LF retains the previous best solution for one player and generates only the remaining 59 players. At steady state, the previous best solution will continue to remain the best solution. Therefore the 59 players have to converge and meet the stopping criteria. This ensures that when the process is at steady state, the MV values remain undisturbed, and the CVs continue to remain at steady state. Additionally, retaining the previous best solution for one of the players reduces the computational burden. When the CVs are not at the setpoints, the previous best solution is no longer the best and LF will optimize the MV moves to find a new solution.

For Case 1 the setpoint was changed at 117 minutes. The controller results are shown in Figure 38. For Case 1 alone, the controller was initially in the Manual mode (open-loop steady state) and transferred to the Auto mode (closed-loop steady state) at 87 minutes. The setpoint is tracking the noisy CV and sets the value of the last sample in the Manual mode as the setpoint for the Auto

Table 16: Setpoint tracking cases

Case	Setpoint change	Tuning	MV change and settling time
1*	X_{top} =0.84 → 0.88 X_{bottom} = 0.02→0.04	τ_{top} = 3 min. τ_{bot} = 3 min. EC_{top} = 1	Reboiler Duty = 205→161 Btu/min Reflux Rate = 0.0063 →0.0047 lbmol/min Settling time = 60 minutes
1	X_{top} =0.84 → 0.88 X_{bottom} = 0.02→0.05	τ_{top} = 1 min. τ_{bot} = 1 min. EC_{top} = 1	Reboiler Duty = 205→161 Btu/min Reflux Rate = 0.0063 →0.0047 lbmol/min Settling time = 40 minutes
2	X_{top} = 0.88 X_{bottom} = 0.05	τ_{top} = 0.2 min. τ_{bot} = 0.2 min. EC_{top} = 1	Excessively aggressive tuning did not allow the MVs to reach steady state.
3	X_{top} = 0.88 X_{bottom} = 0.05	τ_{top} = 0.5 min. τ_{bot} = 0.5 min. EC_{top} = 1	Reboiler Duty = 160 Btu/min Reflux Rate = 0.0046 lbmol/min Settling time = 30 minutes
4	X_{top} = 0.88 X_{bottom} = 0.05	τ_{top} = 3 min. τ_{bot} = 3 min. EC_{top} = 1	Reboiler Duty = 161 Btu/min Reflux Rate = 0.0047 lbmol/min Settling time = 60 minutes
5	X_{top} = 0.88 X_{bottom} = 0.05	τ_{top} = 1 min. τ_{bot} = 1 min. EC_{top} = 0.5	Reboiler Duty = 161 Btu/min Reflux Rate = 0.0047 lbmol/min Settling time = 80 minutes
6	X_{top} = 0.88 X_{bottom} = 0.05	τ_{top} = 1 min. τ_{bot} = 1 min. EC_{top} = 2	Reboiler Duty = 161 Btu/min Reflux Rate = 0.0047 lbmol/min Settling time = 40 minutes

mode. This causes a “bump” in the MVs at 117 minutes (first plot of Figure 38). However, top and bottom tray compositions (the second and third plots of Figure 38) retain their setpoints, thus demonstrating bumpless transfer from Manual to Auto. Immediately after the setpoint change is made, the controller increases the reflux flow rate and decreases the reboiler rate, eventually settling down at a reboiler rate of 161 Btu/min and reflux flow rate of 0.0047 lbmol/min. The controller is thus able to move the process towards the setpoint. The settling time for the process is about 30 minutes for Case 1.

In Case 2 the tuning factors (τ_{top} and τ_{bot}) were set at 0.2 minutes for both the top and bottom tray composition CVs. The setpoint was changed at 53 minutes. The controller results are shown in Figure 39. The controller demonstrates aggressive behavior, with the top tray composition bouncing around the setpoint. The bottom tray composition also demonstrates aggressive behavior but the bottom tray composition is not bouncing around the bottom setpoint, rather it averages slightly below the setpoint. The process shows no signs of settling. Therefore, 0.2 minutes as tuning constant values is unacceptable. For the rest of this work, 1 minute was used as the tuning constant for both τ_{top} and τ_{bot} and an Equal Concern factor of 1 was used for the top composition, unless mentioned otherwise.

In Case 3 the tuning factors (τ_{top} and τ_{bot}) were set at 0.5 minutes for both the top and bottom tray composition CVs. The setpoint was changed at 52 minutes. The controller results are shown in Figure 40. Immediately after the setpoint change is made, the controller increases the reflux flow rate and decreases the reboiler rate, eventually settling down at a reboiler rate of 161 Btu/min and reflux flow rate of 0.0047 lbmol/min. The controller is thus able to move the process towards the setpoint. For Case 3, the settling time for the process is about 30 minutes, which is faster than Case 1 with tuning constant values of 1 minute. Although Case 3 shows slightly aggressive behavior compared to Case 1, the controller is stable.

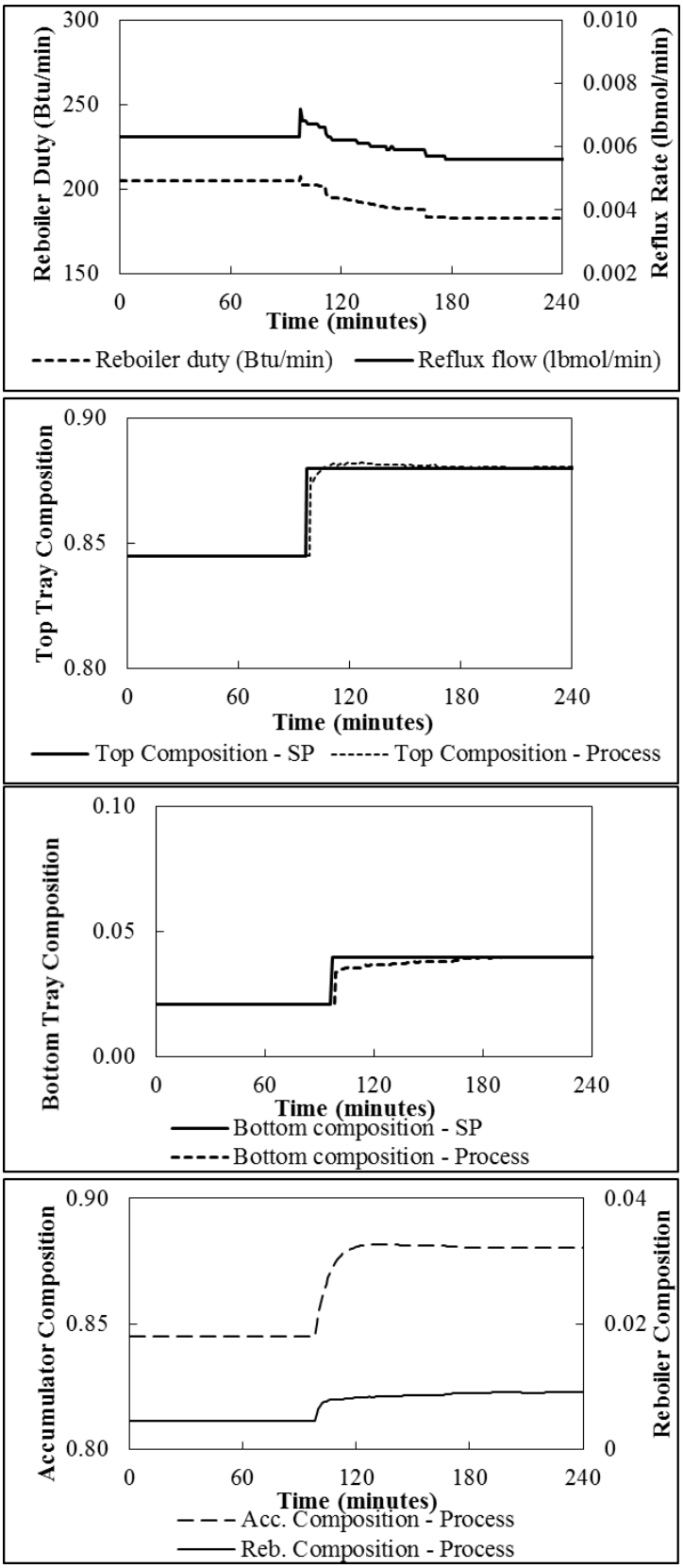


Figure 37: Setpoint tracking (Case 1*)

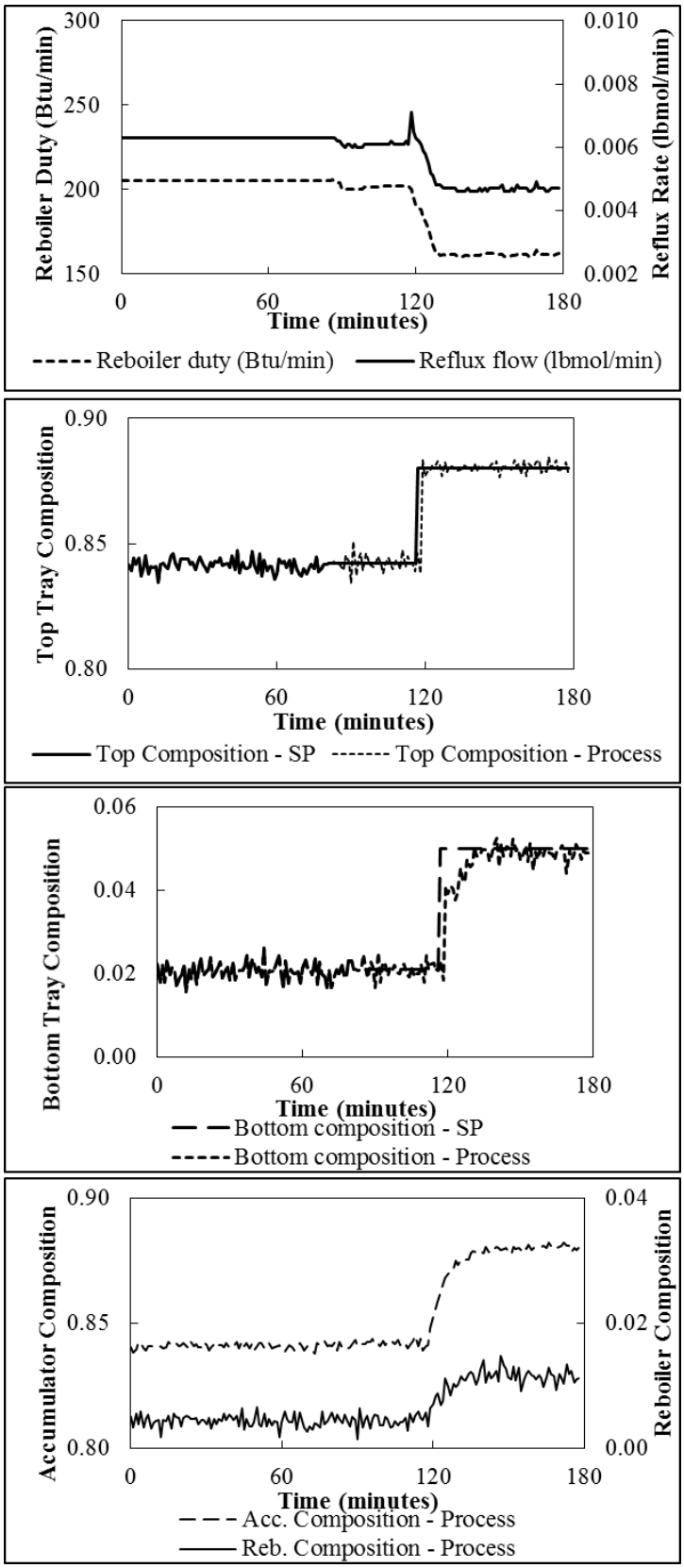


Figure 38: Setpoint tracking (Case 1)

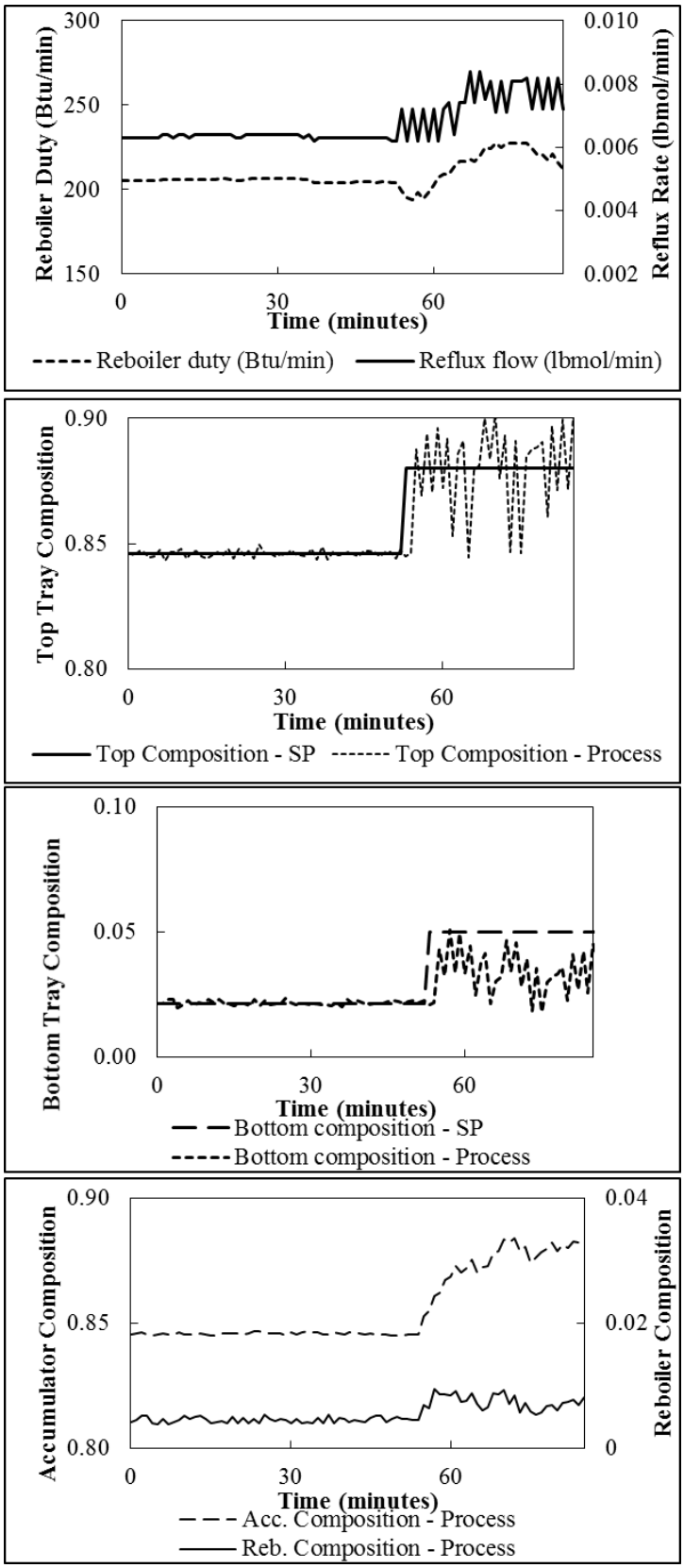


Figure 39: Setpoint tracking (Case 2)

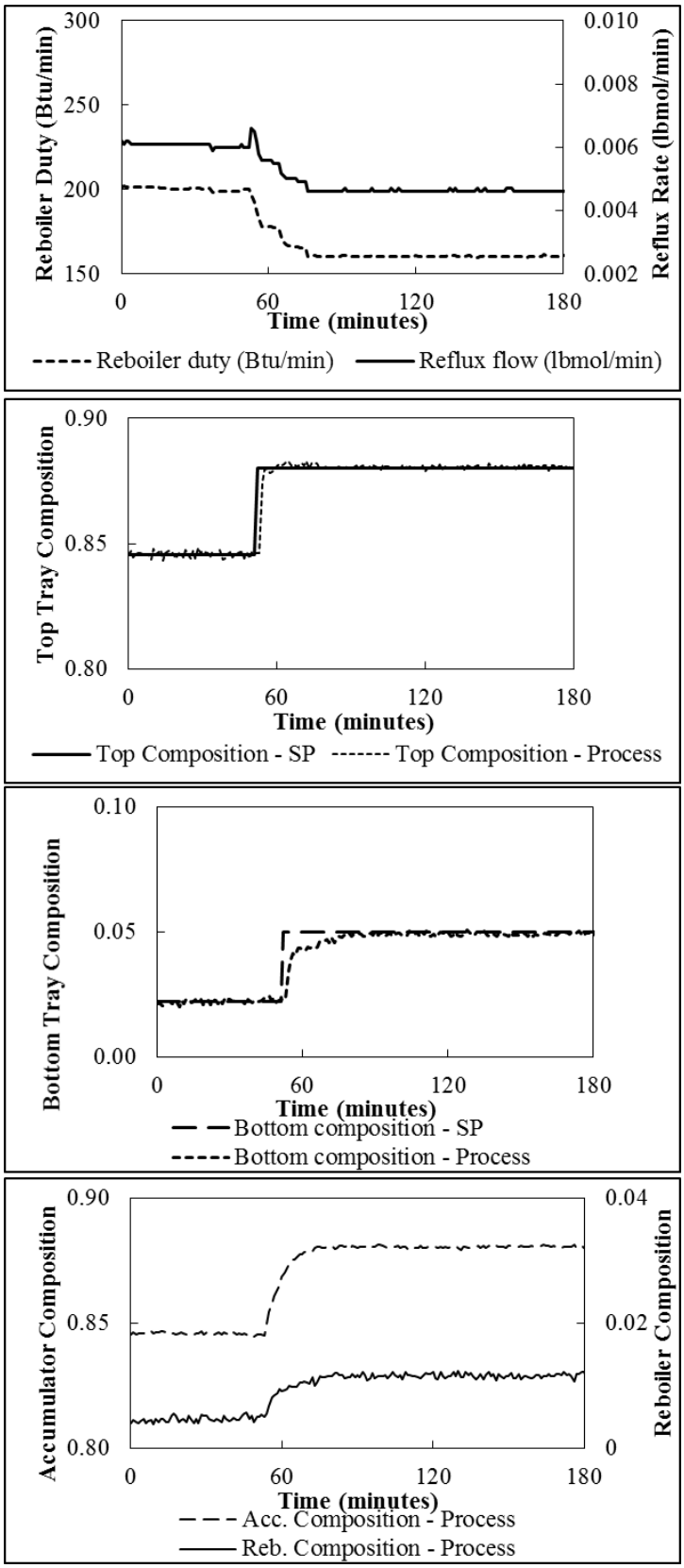


Figure 40: Setpoint tracking (Case 3)

In Case 4 the tuning factors (τ_{top} and τ_{bot}) were set at 3 minutes for both the top and bottom tray composition CVs. The setpoint was changed at 33 minutes. The controller results are shown in Figure 41. Immediately after the setpoint change is made, the controller increases the reflux flow rate and decreases the reboiler rate, eventually settling down at a reboiler rate of 161 Btu/min and reflux flow rate of 0.0047 lbmol/min. The bottom tray composition (third plot of Figure 41) moves slowly and takes a long time to reach the setpoint compared to Cases 1-3. Immediately after the reflux rate increases the top tray composition (second plot of Figure 41) overshoots the setpoint, and takes about 45 minutes to move towards the setpoint. The controller is thus able to move the process towards the setpoint. For Case 3, the settling time for the process is about 30 minutes, which is faster than Case 1 with tuning constant values of 1 minutes. Although Case 3 shows slightly aggressive behavior compared to Case 1, the controller is stable.

In Case 5 the Equal Concern (EC) factor for the top tray composition was set at 0.5 (the EC for the bottom has a default value of 1), meaning that the deviations of the top tray composition was twice as important as the deviations in the bottom tray composition. The setpoint was changed at 44 minutes. The controller results are shown in Figure 42. Immediately after the setpoint change is made, the controller increases the reflux flow rate and decreases the reboiler rate, eventually settling down at a reboiler rate of 161 Btu/min and reflux flow rate of 0.0047 lbmol/min. Immediately after the reflux rate increases the top tray composition (second plot of Figure 42) moves towards the setpoint in about 5 minutes. The accumulator composition (fourth plot of Figure 42) settles at the new steady state in about 20 minutes. The bottom tray composition (third plot of Figure 42) moves slowly and takes a long time to reach the setpoint compared to the top composition because of the lower EC for the top composition. The performance of the controller for Case 5 is as expected, with the top settling down faster than the bottom. Overall, for Case 5, the setting time was about 80 minutes.

In Case 6 the Equal Concern (EC) factor for the top tray composition was set at 2 (the EC for the bottom has a default value of 1), meaning that the deviations of the top tray composition was only half as important as the deviations in the bottom tray composition. The setpoint was changed at 41 minutes. The controller results are shown in Figure 43. Immediately after the setpoint change is made, the controller increases the reflux flow rate and decreases the reboiler rate, eventually settling down at a reboiler rate of 161 Btu/min and reflux flow rate of 0.0047 lbmol/min. Immediately after the reflux rate increases the top tray composition (second plot of Figure 43) overshoots the setpoint and takes about 25 minutes to move towards the steady state. The accumulator composition (fourth plot of Figure 43) settles at the new steady state in about 40 minutes. The bottom tray composition (third plot of Figure 43) moves faster towards the setpoint compared to the top composition because of the higher EC for the top composition. The performance of the controller for Case 6 is as expected, with the bottom settling down faster than the top. Overall, for Case 6, the setting time was about 40 minutes.

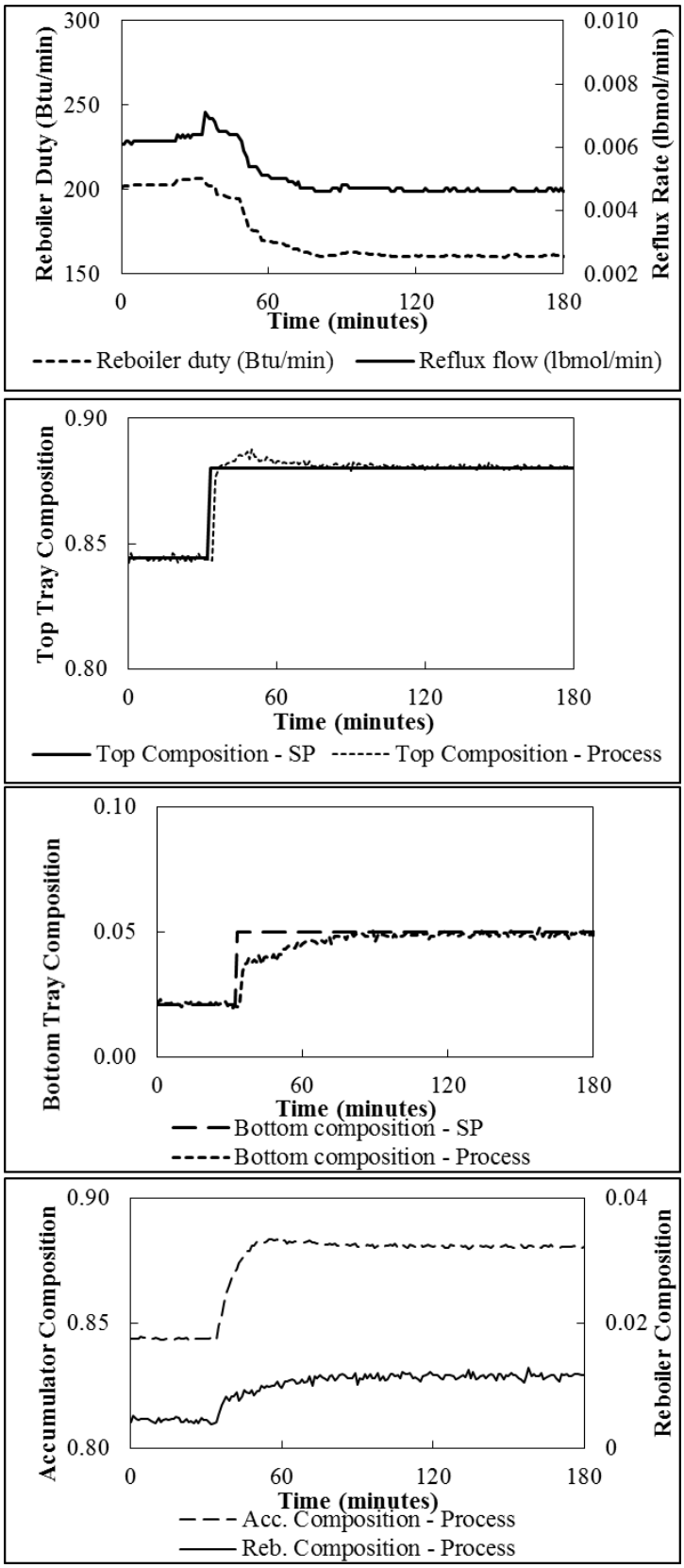


Figure 41: Setpoint tracking (Case 4)

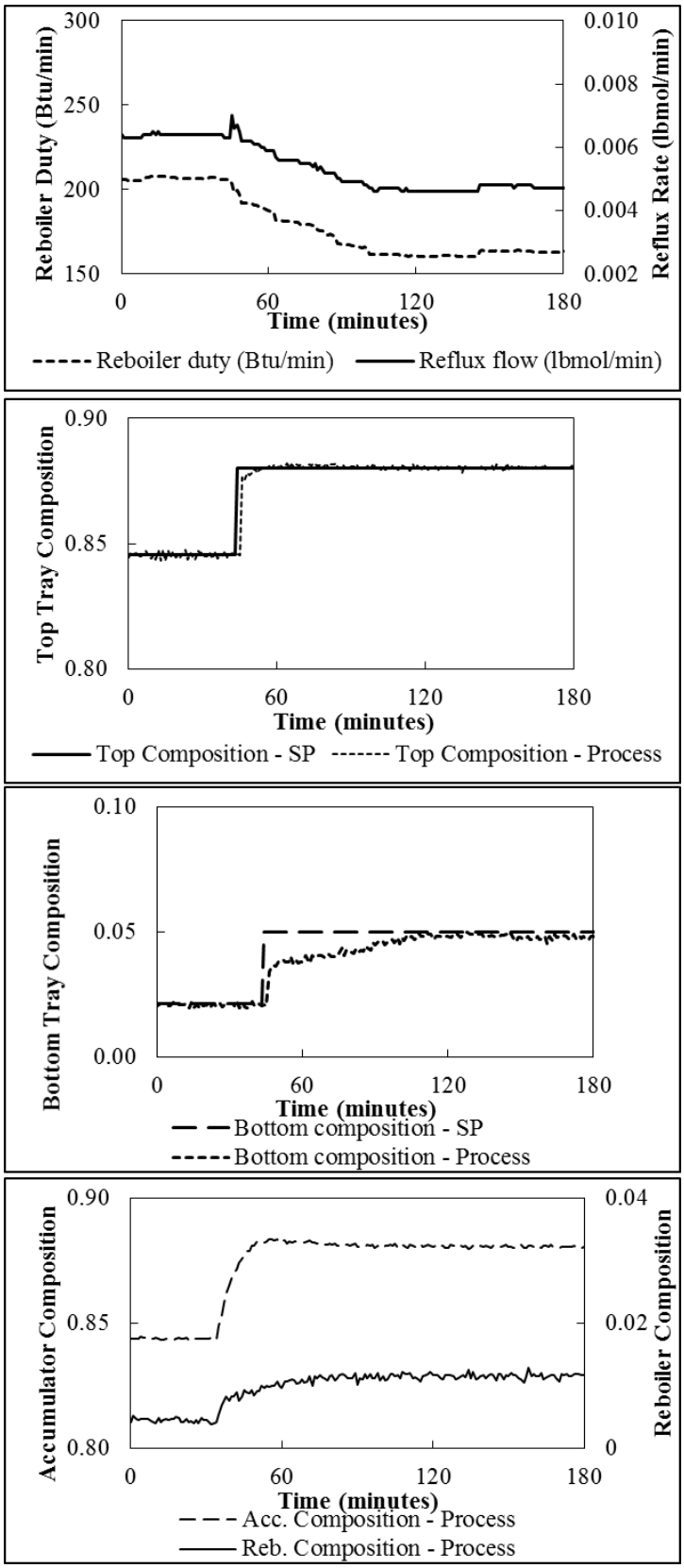


Figure 42: Setpoint tracking (Case 5)

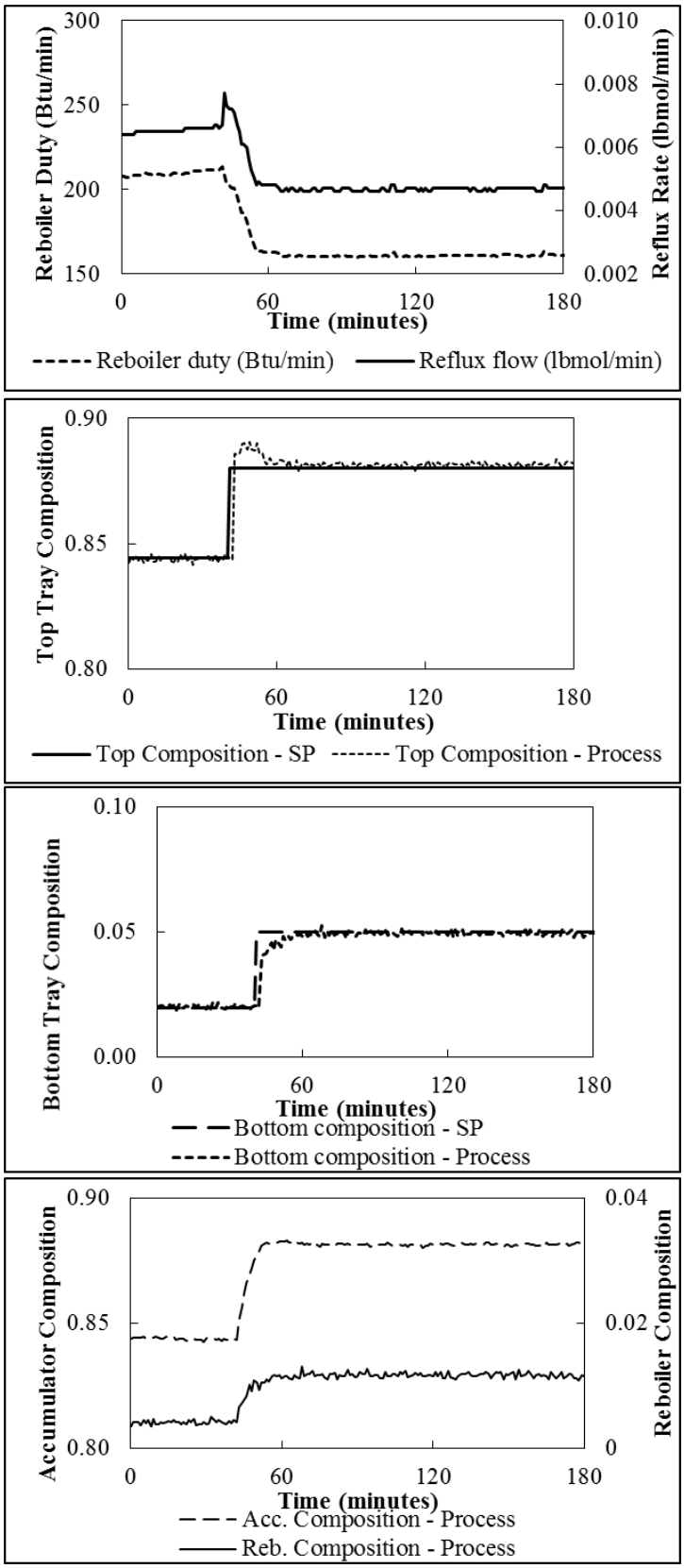


Figure 43: Setpoint tracking (Case 6)

5.4.3 Disturbance rejection

Table 17 lists the disturbance rejection cases that were tested. The disturbance rejection tests were conducted starting with a feed rate of 0.008 lbmol/min and feed composition of 0.5 mole fraction of methanol. In Case 7, the feed rate was changed from an initial rate of 0.008 lbmol/min to 0.009 lbmol/min. The controller results of Case 7 are shown in Figure 44. The disturbance was introduced at 93 minutes. Immediately, the top and bottom tray compositions are upset from their setpoints (second and third plots of Figure 44). The controller responds immediately by increasing the reflux rate and the reboiler duty to counter the change in feed rate and bring the CVs back to their setpoints. Even though the compositions of the top and bottom tray are upset, the accumulator and reboiler compositions remain practically undisturbed (bottom plot of Figure 44).

Table 17: Disturbance rejection cases

Case	Disturbance	Change
7	Feed rate	Initial: 0.008 lbmol/min New: 0.009 lbmol/min
8	Feed composition	Initial: 0.5 mole fraction of methanol New: 0.6 mole fraction of methanol

In Case 8 the feed composition was changed from an initial mole fraction of 0.5 (of methanol) to 0.6 (of methanol). The controller results of Case 8 are shown in Figure 45. The disturbance was introduced at 60 minutes. Immediately after the introduction of the disturbance, the top and bottom tray compositions show sharp deviations from their setpoints (second and third plots of Figure 45). Consequently, the controller adjusts the reflux rates and reboiler duty to move the process back towards the setpoint. While, the change in reflux rate is significant, the change in

reboiler duty is not clearly evident. As in Case 7, even though the compositions of the top and bottom tray are upset, the accumulator and reboiler compositions remain practically undisturbed (bottom plot of Figure 45).

5.4.4 Constraint handling

Table 18 lists the constraint handling cases that were tested. The constraint handling tests were conducted using a feed rate of 0.008 lbmol/min and feed composition of 0.6 mole fraction of methanol. Figure 46 shows the control results of Case 9. In Case 9, the setpoint was changed from 0.90 to 0.96 for the top and 0.05 to 0.10 for the bottom tray composition at 29 minutes. Subsequently, the controller responds by increasing the reflux rate and reboiler duty (top plot of Figure 46). Both the CVs cannot reach their setpoints, with the bottom tray composition (third plot of Figure 46) being closer to the setpoint, compared to the top tray composition (second plot of Figure 46) The controller hits an operating constraint when the reflux rate is 0.015 lbmol/min at about 50 minutes and remained at the constraint until the next setpoint change was made (top plot of Figure 46). The current setpoints were retained until 103 minutes. The controller was not able to move both the CVs towards their setpoints because it hit the upper limit of the operating constraint for reflux rate. At 103 minutes when the setpoints were changed to 0.93 for the top and 0.08 for the bottom tray composition, the controller immediately relieved the reflux MV from its constraint and brought the CVs to the setpoints. When the setpoint change was made at 103 minutes, the reboiler duty increases at first to compensate for the high reflux rate before it reverses direction and drops to lower duties (top plot of Figure 46).

Table 18: Constraint handling cases

Case	Setpoint change	Initial steady state MVs	Constraint hit
9	X_{top} =0.90 → 0.96→0.93 X_{bottom} = 0.05→0.1→0.08	Reboiler Duty = 205 Reflux Rate = 0.0063	Reflux rate = 0.015 lbmol/min (upper limit)
10	X_{top} =0.915 → 0.84→0.83 X_{bottom} = 0.063→0.04→0.03	Reboiler Duty = 205 Reflux Rate = 0.0063	Reboiler duty = 160 Btu/min (lower limit)

Figure 47 shows the control results of Case 10. In Case 10, the setpoint was at 29 minutes. Subsequently, the controller responds by decreasing the reflux rate and reboiler duty (top plot of Figure 47). Neither the top tray, nor the bottom tray composition get to their setpoints (second and third plot of Figure 47) because the controller hits an operating constraint when the reboiler duty is 160 Btu/min at about 70 minutes. The current setpoints were retained until 110 minutes. At 110 minutes when the setpoints were changed to 0.83 for the top and 0.03 for the bottom tray composition, the controller immediately relieved the reflux MV from its constraint and brought the CVs to the setpoints.

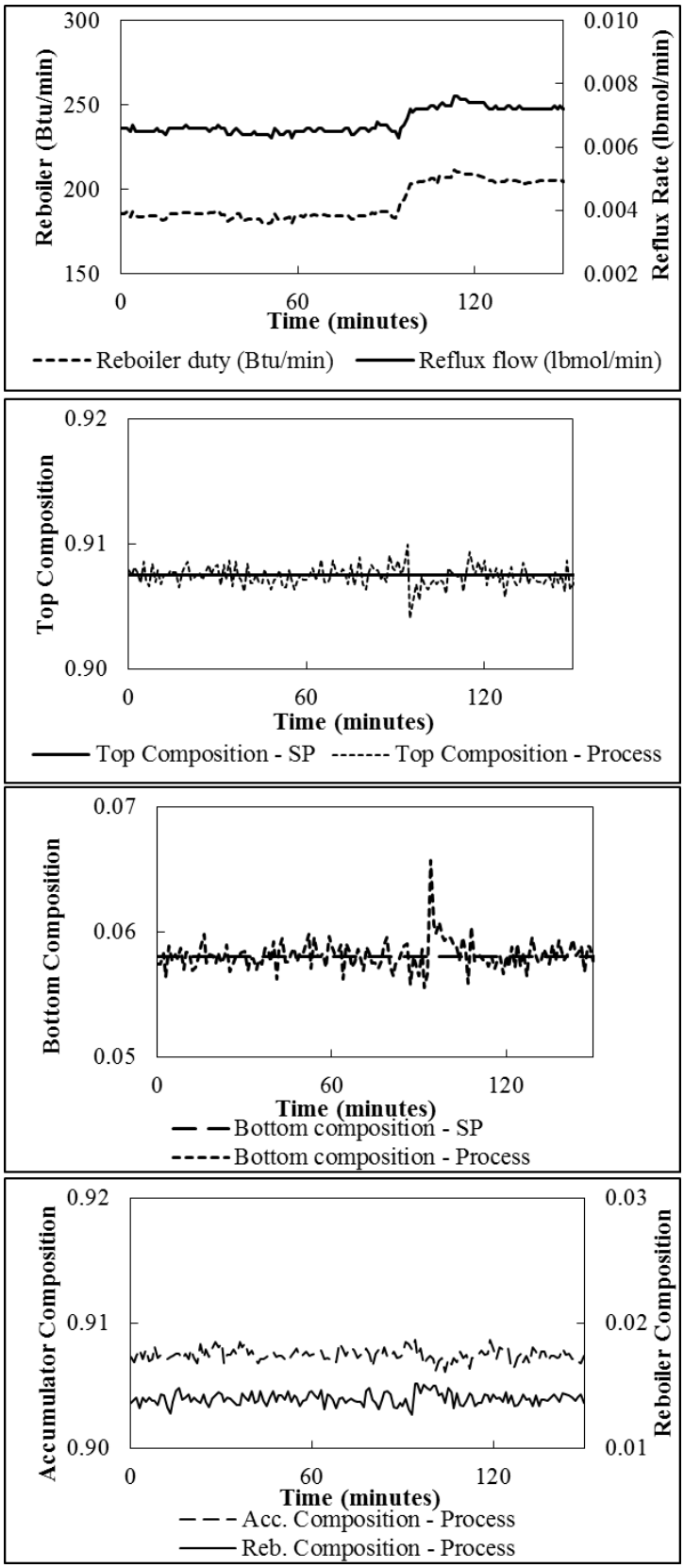


Figure 44: Disturbance rejection (Case 7)

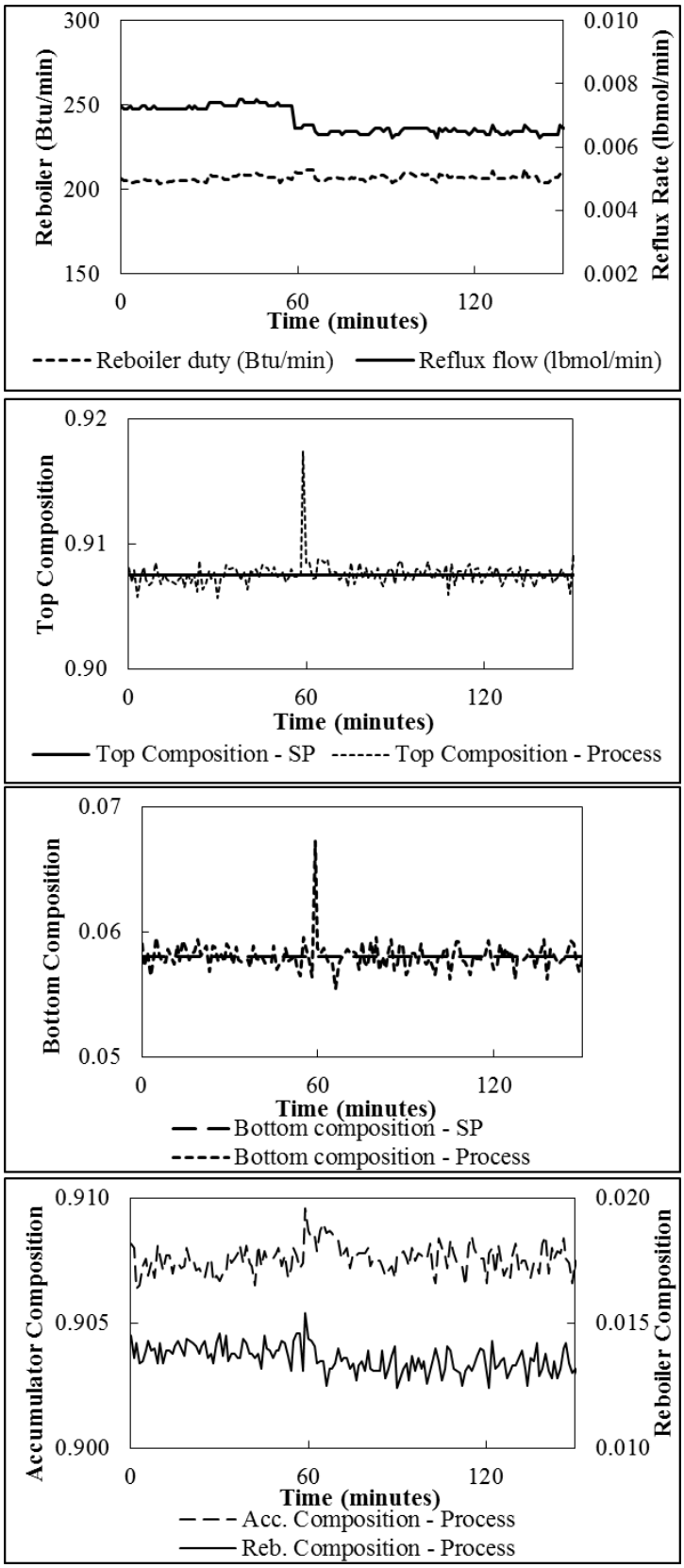


Figure 45: Disturbance rejection (Case 8)

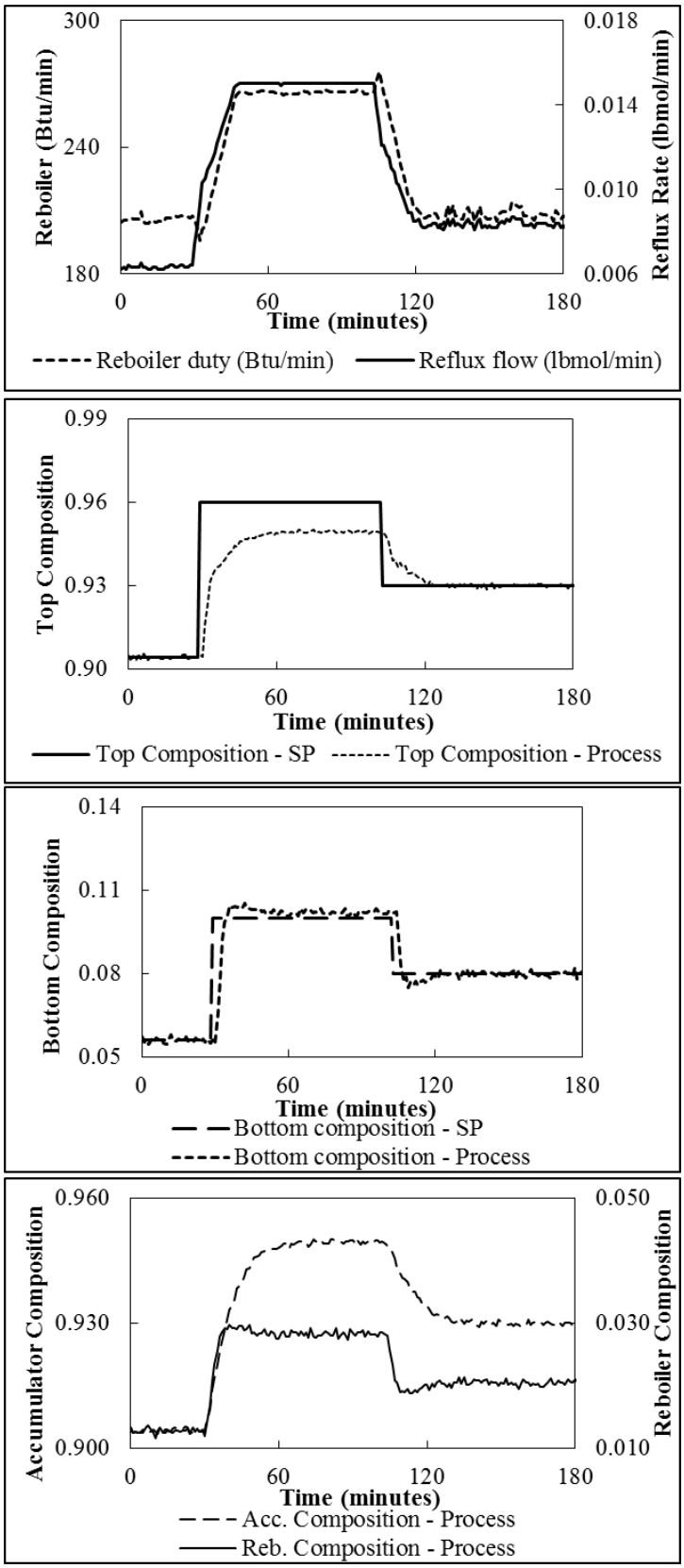


Figure 46: Constraint handling (Case 9)

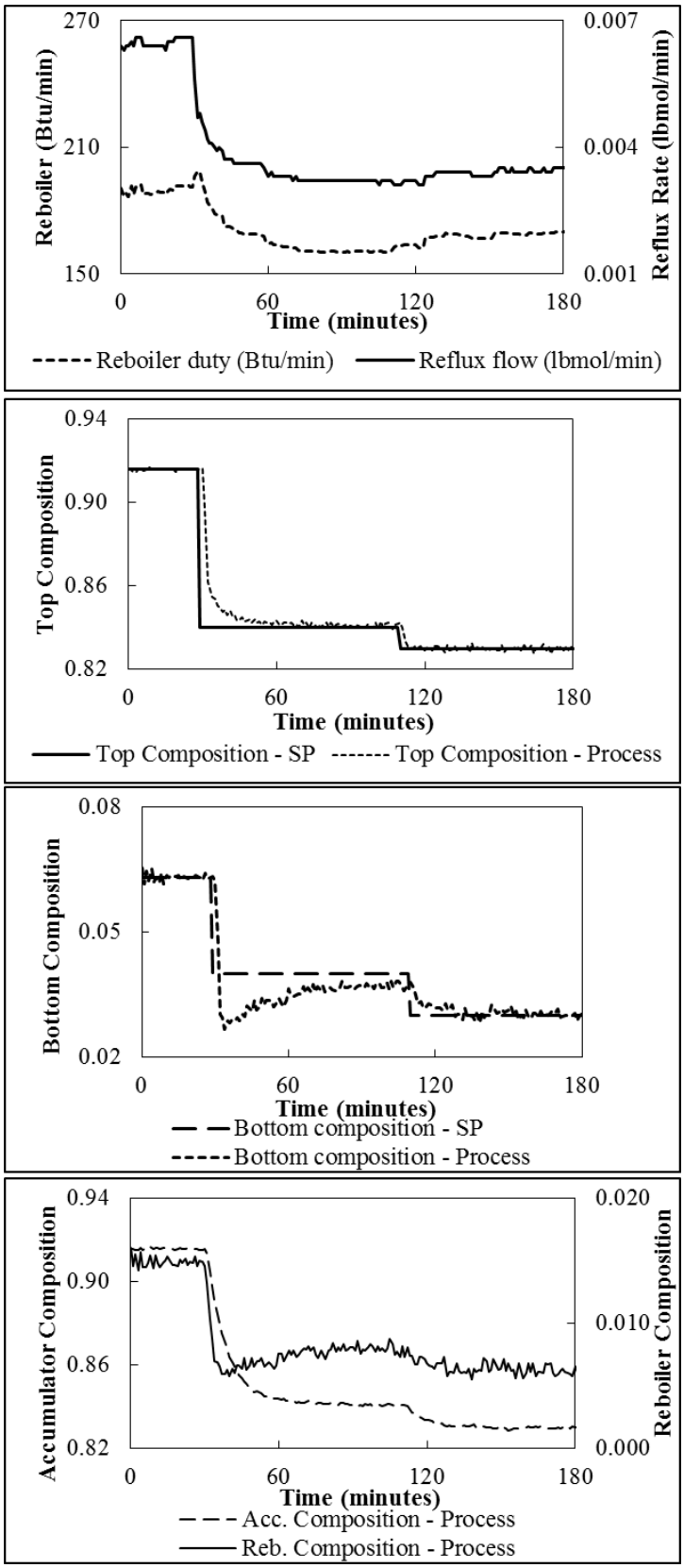


Figure 47: Constraint handling (Case 10)

CHAPTER VI

CONCLUSIONS AND FUTURE WORK

6.1 Accelerated convergence improvement

In this work, methods to accelerate convergence of LF optimization by modifying the leap-to window size factor α, were explored and analyzed. Based on the test simulations, the limited variety of test functions chosen, the ΔDV convergence criteria and the number of players used per dimension, the following conclusions were drawn regarding *α*, the leap-to window size factor,

- The number of leapovers to convergence after locating the vicinity of the global, or the number of leapovers to convergence (in case of single optimum) is a linear function of the negative reciprocal of ln(0.5* α).
- When the players are initialized in a DV range that encompasses the global, for 6 out of 9 test cases, α = 0.5 provides the best balance of reduction in number of leapovers (speed of computation) and maintaining a similar level of PNOFE compared to α = 1.0 (robustness).
- When the players are initialized locally, for 5 out of 9 test cases, α = 1.0 provides the best balance of reduction in number of leapovers while maintaining PNOFE

- The mathematical analysis concerning leap-to window size factors and the subsequent analysis presented in this work provides fundamental understanding of LF, which is important to create opportunities for algorithm improvement.

6.2 LF and nonlinear process modeling

Concerning LF in process modeling applications such as binary distillation the following conclusions were drawn,

- LF is a useful method to solve the optimization model of a nonlinear, steady state or dynamic, first principles distillation model with interacting variables that closed the material and energy balances.
- LF can find solutions to problems with varying DV scales. In the case of binary distillation modeling, the liquid mole fraction DVs have a range of 0 to 1, while the molar liquid flow rate DVs have a range of Reflux rate to 2*Reflux rate + Feed rate (numerically from 0.001 lbmol/min to 0.045 lbmol/min). LF does not require any scaling factors to handle the different DV ranges.
- LF can handle interacting DVs. For instance, a change on the molar liquid mole flow on a stage affects the liquid mole fraction on all other stages. The ability to handle interacting DVs is important to several chemical engineering applications such as distillation, and absorption.
- LF can find solutions to problems where the range (difference between the best and worst values) at initialization is several orders of magnitude higher than the range at stopping.

Although not explored in this study, prior studies revealed that LF finds an optimum with fewer NOFE than classical gradient-based or direct search optimizers [2]. Coupled with that, to a practicing engineer, this demonstration of LF on nonlinear process modeling applications such as distillation provides credibility and proof-of-concept.

6.3 LF and NMPC

Demonstrating application of LF to NMPC of a binary distillation simulation creates application credibility for process control engineers who could be interested in using first principles models for process control. Additionally, based on the development and testing of LF on NMPC of binary distillation one can conclude that LF can handle,

- Multivariable control with nonlinear, interacting MVs
- Severe disturbances
- MV constraints
- Rate of change constraints

6.4 Summary

Accelerating convergence of LF and demonstrating applications on nonlinear process modeling and NMPC pave the way for testing applications to larger scale problems such as real time optimizers and refinery planning, which involve several hundred variables and where time and computational burden are key.

6.5 Future work

LF has proven applications and advantages, and is a potential best-in-class method [2, 17, 34, 36, 40, 43]. However, there are significant opportunities for future work on algorithm improvements and developing new relevant applications.

- Start with improved initialization and proceed with optimization using a smaller leap-to window size factor (α) based on the recommendations of this work. Improved initialization starts with many individuals to increase the probability of finding the global, and then selects only the best subset of players for optimization [43]. Improved

initialization gives LF a head start and provides a high confidence that at least one player is in the vicinity of the optimum. Therefore, when improved initialization is combined with the lower leap-to window size factor, the expectation is that LF will take fewer leapovers to reach the optimum.

- Based on the experience of this work and several others in the past, it appears that the first several leapovers of LF (on average a fifth of the total leapovers) primarily contribute to exploration (and thus finding the vicinity of the global), and the remaining four fifths of leapovers are expended in trying to converge at the solution. An understanding of this progression of work will help improve the algorithm. Perhaps, start with a larger leap-to window size during the initial exploration phase, and reduce to a smaller leap-to window size to hasten convergence.
- Implement the NMPC simulation of binary distillation demonstrated in this work on the UOL distillation column. This work mimics reality by adding noise to the process model. The reboiler and accumulator liquid volumes are assumed to be smaller than the actual volumes in the UOL distillation column to enhance the speed of the simulation. When NMPC is used as a supervisory controller sending setpoints to existing regulatory controllers (which are typically linear such as PI) additional non-idealities will result such as the valve and reboiler dynamics. The model is only a reasonable representation of the process, and is not exactly true, therefore it is important to test the validity of this simulation on NMPC of distillation.
- Use LF to develop RTO application simulations. Real time optimization is carried out to maximize economic benefit, and to improve resource utilization of an operating plant. Real time optimization models are computed only once every hour or several hours because of their complexity. Showing application of LF to RTO applications with several hundred variables will further enhance credibility of LF.

REFERENCES

1. Lee, J., *Model Predictive Control: Review of the Three Decades of Development.* International Journal of Control, Automation and Systems, 2011. **9**(3): p. 415-424.
2. Rhinehart, R.R., M. Su, and U. Manimegalai-Sridhar, *Leapfrogging and Synoptic Leapfrogging: A New Optimization Approach.* Computers & Chemical Engineering, 2012. **40**(0): p. 67-81.
3. Babu, B.V. and R. Angira, *Modified Differential Evolution (Mde) for Optimization of Non-Linear Chemical Processes.* Computers & Chemical Engineering, 2006. **30**(6–7): p. 989-1002.
4. Golshan, M., M.R. Pishvaie, and R. Bozorgmehry Boozarjomehry, *Stochastic and Global Real Time Optimization of Tennessee Eastman Challenge Problem.* Engineering Applications of Artificial Intelligence, 2008. **21**(2): p. 215-228.
5. Hsu, C.-C. and G.-Y. Lin, *Digital Redesign of Uncertain Interval Systems Based on Time-Response Resemblance Via Particle Swarm Optimization.* ISA Transactions, 2009. **48**(3): p. 264-272.
6. van der Lee, J.H., W.Y. Svrcek, and B.R. Young, *A Tuning Algorithm for Model Predictive Controllers Based on Genetic Algorithms and Fuzzy Decision Making.* ISA Transactions, 2008. **47**(1): p. 53-59.
7. Li, J. and R. Russell Rhinehart, *Heuristic Random Optimization.* Computers & Chemical Engineering, 1998. **22**(3): p. 427-444.

8. Ochoa, S., G. Wozny, and J.-U. Repke, *A New Algorithm for Global Optimization: Molecular-Inspired Parallel Tempering.* Computers & Chemical Engineering, 2010. **34**(12): p. 2072-2084.

9. Ramezani, M.H. and N. Sadati, *Hierarchical Optimal Control of Large-Scale Nonlinear Chemical Processes.* ISA Transactions, 2009. **48**(1): p. 38-47.

10. Singh, V., I. Gupta, and H.O. Gupta, *Ann Based Estimator for Distillation—Inferential Control.* Chemical Engineering and Processing: Process Intensification, 2005. **44**(7): p. 785-795.

11. Nelder, J.A. and R. Mead, *A Simplex Method for Function Minimization.* The Computer Journal, 1965. **7**(4): p. 308-313.

12. Hooke, R. and T.A. Jeeves, ``*Direct Search''Solution of Numerical and Statistical Problems.* Journal of the ACM (JACM), 1961. **8**(2): p. 212-229.

13. Moré, J., *The Levenberg-Marquardt Algorithm: Implementation and Theory*, in *Numerical Analysis*, G.A. Watson, Editor. 1978, Springer Berlin Heidelberg. p. 105-116.

14. Glover, F. and H. Greenberg, *Introduction.* Annals of Operations Research, 1989. **21**(1): p. ix-xii.

15. Agrawal, V., H. Sharma, and J. Bansal, *Bacterial Foraging Optimization: A Survey*, in *Proceedings of the International Conference on Soft Computing for Problem Solving (Socpros 2011) December 20-22, 2011*, K. Deep, A. Nagar, M. Pant, and J.C. Bansal, Editors. 2012, Springer India. p. 227-242.

16. Allgower, F., R. Findeisen, and Z.K. Nagy, *Nonlinear Model Predictive Control: From Theory to Application.* Chinese Institute of Chemical Engineers, 2004. **35**(3): p. 299-316.

17. Haoxian, C. and R.R. Rhinehart. *Incorporation of a Generalized Tsk Model in Nonlinear Model Predictive Control*. in *American Control Conference (ACC), 2012*. 2012.

18. Rhinehart, R.R. *Tutorial: Process Control through Nonlinear Modeling*. in *American Control Conference, 1997. Proceedings of the 1997*. 1997.

19. Canney, W.M., *The Future of Advanced Process Control Promises More Benefits and Sustained Value.* Oil & gas journal, 2003. **101**(16): p. 48-54.

20. Morari, M. and J. H. Lee, *Model Predictive Control: Past, Present and Future.* Computers & Chemical Engineering, 1999. **23**(4–5): p. 667-682.

21. Conway, B., *A Survey of Methods Available for the Numerical Optimization of Continuous Dynamic Systems.* Journal of Optimization Theory and Applications, 2012. **152**(2): p. 271-306.

22. Qin, S.J. and T.A. Badgwell, *A Survey of Industrial Model Predictive Control Technology.* Control Engineering Practice, 2003. **11**(7): p. 733-764.

23. Cutler, C.R. and B. Ramaker. *Dynamic Matrix Control-a Computer Control Algorithm*. in *Proceedings of the joint automatic control conference*. 1980.

24. Bradley Anderson, A.B., Michael Salciccioli. *Model Predictive Control*. 2007 10/30/2013]; Available from: http://controls.engin.umich.edu/wiki/index.php/MPC.

25. Young, R.E., R.D. Bartusiak, and R.W. Fontaine. *Evolution of an Industrial Nonlinear Model Predictive Controller*. in *AICHE SYMPOSIUM SERIES*. 2002. New York; American Institute of Chemical Engineers; 1998.

26. Bauer, M. and I.K. Craig, *Economic Assessment of Advanced Process Control–a Survey and Framework.* Journal of Process Control, 2008. **18**(1): p. 2-18.

27. Alessio, A. and A. Bemporad, *A Survey on Explicit Model Predictive Control*, in *Nonlinear Model Predictive Control*, L. Magni, D. Raimondo, and F. Allgöwer, Editors. 2009, Springer Berlin Heidelberg. p. 345-369.

28. Froisy, J.B., *Model Predictive Control—Building a Bridge between Theory and Practice.* Computers & Chemical Engineering, 2006. **30**(10–12): p. 1426-1435.

29. Garcia, C.E. and A. Morshedi, *Quadratic Programming Solution of Dynamic Matrix Control (Qdmc).* Chemical Engineering Communications, 1986. **46**(1-3): p. 73-87.

30. Lucia, A. and J. Xu, *Methods of Successive Quadratic Programming.* Computers & Chemical Engineering, 1994. **18, Supplement 1**(0): p. S211-S215.

31. Han, S.-P., *Superlinearly Convergent Variable Metric Algorithms for General Nonlinear Programming Problems.* Mathematical Programming, 1976. **11**(1): p. 263-282.

32. Powell, M.J.D., *A Fast Algorithm for Nonlinearly Constrained Optimization Calculations*, in *Numerical Analysis*, G.A. Watson, Editor. 1978, Springer Berlin Heidelberg. p. 144-157.

33. Iyer, M.S. and R.R. Rhinehart, *A Method to Determine the Required Number of Neural-Network Training Repetitions.* Neural Networks, IEEE Transactions on, 1999. **10**(2): p. 427-432.

34. Ratakonda, S., U.M. Sridhar, R.R. Rhinehart, and S.V. Madihally, *Assessing Viscoelastic Properties of Chitosan Scaffolds and Validation with Cyclical Tests.* Acta Biomaterialia, 2012. **8**(4): p. 1566-1575.

35. Sethuraman, V., K. Makornkaewkeyoon, A. Khalf, and S.V. Madihally, *Influence of Scaffold Forming Techniques on Stress Relaxation Behavior of Polycaprolactone Scaffolds.* Journal of Applied Polymer Science, 2013. **130**(6): p. 4237-4244.

36. Raul, P.R., H. Srinivasan, S. Kulkarni, M. Shokrian, G. Shrivastava, and R. Russell Rhinehart, *Comparison of Model-Based and Conventional Controllers on a Pilot-Scale Heat Exchanger.* ISA Transactions, 2013. **52**(3): p. 391-405.

37. Manimegalai Sridhar, U., *Pseudo-Component Viscoelastic Modeling of Soft Tissues*, in *Chemical Engineering*. 2010, Oklahoma State University: Stillwater, OK.

38. Ratakonda, S., *Assessing Viscoelastic Properties of Chitosan Scaffolds and Validating Sequential and Cyclical Tests*, in *Chemical Engineering*. 2011, Oklahoma State University: Stillwater, OK.

39. Chen, H., *Incorporation of the Generalized Tsk Models in Model Predictive Control*, in *Chemical Engineering*. 2011, Oklahoma State University: Stillwater, OK.

40. Rhinehart, R.R., *Convergence Criterion in Optimization of Stochastic Processes.* Computers & Chemical Engineering, 2014. **68**(0): p. 1-6.

41. Manimegalai-Sridhar, U., A. Govindarajan, and R. Russell Rhinehart, *Improved Initialization of Players in Leapfrogging Optimization.* Computers & Chemical Engineering, (0).

42. Seader, J., *Separation Process Principles*. 2006: Wiley. com.

43. Upasana Manimegalai-Sridhar, A.G., R. Russell Rhinehart, *Improved Initialization of Players in Leapfrogging Optimization.* Computers & Chemical Engineering, 2013. **In Press**.

44. Box, G.E.P. and M.E. Muller, *A Note on the Generation of Random Normal Deviates.* 1958(2): p. 610-611.

APPENDIX A

OPTIMIZATION TEST FUNCTIONS

This appendix provides the computer codes used for generating the optimization test functions used in this dissertation.

Function F1 – Peaks

```
x11 = 3 * (x1 - 5) / 5      'convert 0-10 DV scale to the -3 to +3 range for the function
x22 = 3 * (x2 - 5) / 5
f_of_x = 3 * ((1 - x11) ^ 2) * Exp(-1 * x11 ^ 2 - (x22 + 1) ^ 2) - 10 * (x11 / 5 - x11 ^ 3 - x22 ^ 5)
*Exp(-1 * x11 ^ 2 - x22 ^ 2) - (Exp(-1 * (x11 + 1) ^ 2 - x22 ^ 2)) / 3
f_of_x = (f_of_x + 6.75) / 1.5 'scaled for 0-10 f-range
```

Function F2 – Boot Print with Pinhole

```
x1line = 1 + 0.2 * (x2 - 4) ^ 2
deviation = (x1line - x1)
penalty = 5 * (1 / (1 + Exp(-3 * deviation)))    'logit functionality
f_of_x = 0.5 * x1 - 0.2 * x2 + penalty + add_noise
x1mc2 = (x1 - 1.5) ^ 2
x2mc2 = (x2 - 8.5) ^ 2
factor = 1 + (5 * (x1mc2 + x2mc2) - 2) * Exp(-4 * (x1mc2 + x2mc2))
f_of_x = factor * f_of_x
f_of_x = 10 * (f_of_x - 0.3) / 6
```

Function F3 – Goldstein and Price

o1 = 0.4 * x1 - 2

o2 = 0.4 * x2 - 2

f_of_x = (1 + ((o1 + o2 + 1) ^ 2) * (19 - 14 * o1 + 3 * o1 ^ 2 - 14 * o2 + 6 * o1 * o2 + 3 * o2 ^ 2)) * (30 + ((2 * o1 - 3 * o2) ^ 2) * (18 - 32 * o1 + 12 * o1 ^ 2 + 48 * o2 - 36 * o1 * o2 + 27 * o2 ^ 2))

f_of_x = 2 * Sqr((f_of_x + 1) / 8000)

Function F4 – Simple Ellipse

f_of_x = 0.1 * (3 * (x1 - 5) ^ 2 + (x2 - 6) ^ 2)

Function F5 – Hot and Cold Water Mixing MPC

o1 = 10 * x1'hot valve position, %

o2 = 10 * x2'cold valve position, %

SetpointT = 70'Celsius

FromT = 35'Celsius

SetpointF = 20'm^3/min

FromF = 8'm^3/min

HotTin = 80'Celsius

ColdTin = 20 * (1 + add_noise)'Celsius

ValveCv = 0.0036 * (1 + add_noise)'m^3/min/%^2

EC4T = 0.15'Celsius^(-2)

EC4F = 1'(m^3/min)^(-2)

f_of_x = EC4T * (1.2 * (SetpointT - FromT) + FromT - (HotTin * o1 ^ 2 + ColdTin * o2 ^ 2) / (o1 ^ 2 + o2 ^ 2)) ^ 2 + EC4F * (1.2 * (SetpointF - FromF) + FromF - ValveCv * (o1 ^ 2 + o2 ^ 2)) ^ 2

f_of_x = f_of_x / 150'scaled for display

Function F6 – 1- Tray Distillation Column

```
FFR = 5 'Feed flow rate moles/time
zF = 0.2 'feed composition, mole fraction
TF = 50 'T in centigrade
rR = 3 'reflux ratio, reflux rate to distillate rate
rB = 2 'boil-up ratio, VB to B
L1FR = (x1 / 2.5) * FFR 'optimizer guess of liquid rate leaving the feed tray
xtray = x2 / 10 'optimizer guess of liquid composition leaving the feed tray
If L1FR < 0 Or xtray < 0 Or xtray > 1 Then
constraint = "Fail"
f_of_x = 100
Exit Function
End If
BFR = L1FR / (1 + rB) 'Boil-up flow rate
DFR = FFR – BFR 'distillate flow rate
VBFR = L1FR – BFR 'boil-up vapor rate
L0FR = rR * DFR 'reflux flow rate
V1FR = (1 + rR) * DFR 'vapor rate exiting the column
If DFR < 0 Or VBFR < 0 Or L0FR < 0 Or V1FR < 0 Then
constraint = "Fail"
f_of_x = 100
Exit Function
End If
T1 = 100 * Exp(-0.233143551 * xtray ^ 0.8) 'equilibrium roll_time for tray 1
y1 = ((100 - T1) / 20) ^ 0.3 'equilibruim y for tray 1
T0 = T1 – 30 'reflux T, sub-cooled from condensor
```

```
xb = xtray 'bottoms liquid composition in a total reboiler

TB = 100 * Exp(-0.233143551 * xb) 'reboiler T

yb = xb 'vapor biol-up composition

x0 = y1  'reflux composition

xd = y1 'distillate composition

hF = zF * 1 * TF + (1 - zF) * 2 * TF 'liquid with reference at T=0

hL0 = x0 * 1 * TL0 + (1 - x0) * 2 * TL0

hL1 = x1 * 1 * TL1 + (1 - x1) * 2 * TL1

HVB = yb * (0.5 * TB + 500) + (1 - yb) * (1 * TB + 2000) 'Vaporized at T=0

HV1 = y1 * (0.5 * T1 + 500) + (1 - y1) * (1 * T1 + 2000)

mass = (FFR * zF + L0FR * x0 + VBFR * yb) - (L1FR * xtray + V1FR * y1)

energy = (FFR * hF + L0FR * hL0 + VBFR * HVB) - (L1FR * hL1 + V1FR * HV1)

f_of_x = 2.5 * Sqr((mass) ^ 2 + (energy / 10000) ^ 2)
```

Function F7 – Sharp Troughs

```
f_of_x = 0.02 * (((x1 - 8) ^ 2 + (x2 - 6) ^ 2) + 15 * Abs((x1 - 2) * (x2 - 4)) - 400 * Exp(-((x1 - 9) ^ 2 + (x2 - 9) ^ 2)))
```

Function F8 – Jupitor's Eye

```
x11 = x1 + 1

x22 = x2 – 1

f_of_x = 3.5 * Log(1 + ((Abs(x11 - 4 - 0.006 * x22 ^ 3)) ^ 2.8 + (Abs(x22 - 6)) ^ 1.2) ^ 0.5)
```

Function F9 – Steady State 11 Stage Distillation Column

```
float x[100], L[100], y[100], Hy[100], hx[100], Temp[100], V[100], F[100];

float dev[100], totdev[100];

float PdtD[100], PdtB[100], Cpm, Cpw, Qc, hf, TeqF, Hvapwater, Hvapmeth, Hreb,hxc ;

float xb, xd, RR, xc, Top, Bottom;

float sumdev1,OFnew;

int dv, N,i,  playperdv, numdv, numpl;
```

```
int  xdv, ldv, constraint, constraint2;

Cpm=0.6; Cpw=1; F[Feedstg]=Feed;

Hvapwater=17525; Hvapmeth=15207;//Btu/lbmol

for (N=0; N<=Nstg-1; N++){

x[N+1]=pX[N];

L[N+1]=pL[N];

if (pX[N]<0|| pL[N]<0){

constraint=constraint+1;}}

Hreb=xbn*Hvapmeth+(1-xbn)*Hvapwater; //Assume simple mixing

V[Nstg-1]=Qr/Hreb; //Vapor boil up

PdtB[Nstg-1] = L[Nstg-2]-V[Nstg-1]; //Based on material balance

if (PdtB[Nstg-1]<0) {PdtB[Nstg-1]=0;}

PdtD[0]=Feed - PdtB[Nstg-1];

if (PdtD[0]<0) {PdtD[0]=0;}

Bottom=PdtB[Nstg-1]; Top=PdtD[0];

L[0]=Reflux;

if (Bottom>=0.9999*Feed|| Top >=0.9999*Feed){

constraint=constraint+1;}

TeqF=-236.53 * Z** 5 + 726.38 * Z **4 - 855.63 * Z ** 3 + 490.17 * Z ** 2 - 159.61 *Z+
99.509; // Equilibrium temperature

hf=(-0.0167 * TeqF** 3 + 4.0511 * TeqF** 2 - 293.95 * TeqF + 8850.2)-((Z*Cpm)+(1-
Z)*Cpw)*((TeqF*1.8+32)-(Tfeed*1.8+32)); // Adjust for subcooling

for (N=0; N<=Nstg-1; N++){

if (constraint>0){

break;}

if (N==0){

V[N] = 0;
```

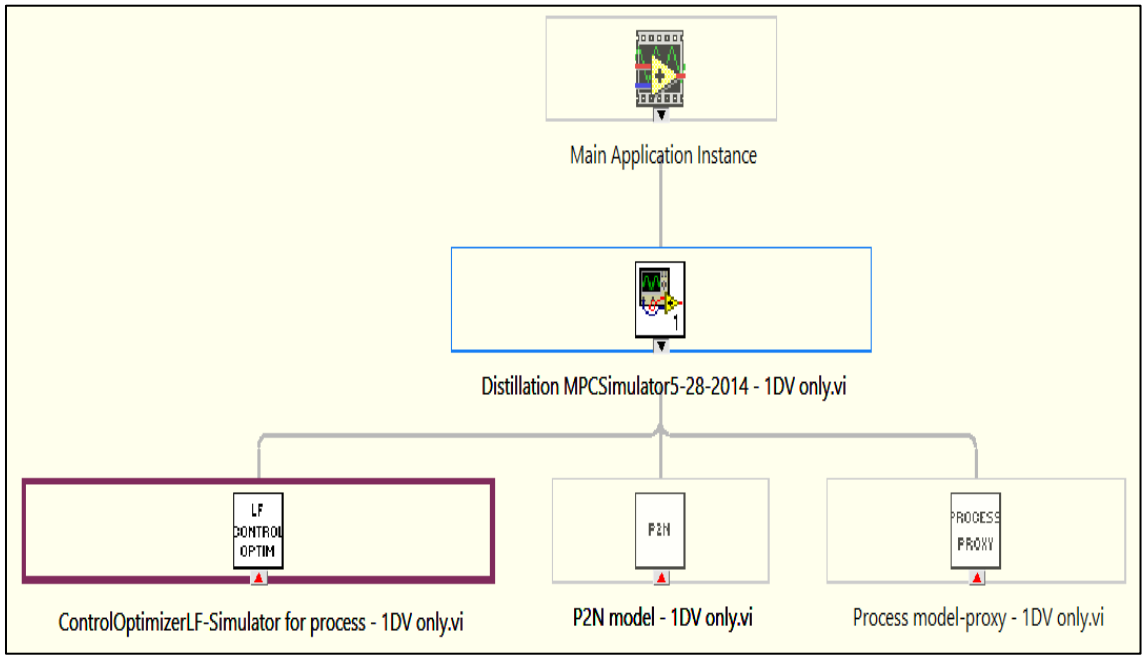
```
y[N]=0;

RR=L[N]/PdtD[N];

y[N+1]= 11.21 * x[N+1] ** 5 - 33.47 * x[N+1] ** 4 + 37.88 * x[N+1]** 3 - 20.28 * x[N+1]** 2 + 5.65 * x[N+1]; //Equilibrium x-y relation

xc=y[N+1]; //Condensate exits as saturated liquid

xd=xc;

x[N]=xdn;

if(xd<Z){constraint=constraint+1;}

Temp[N+1]=-236.53 * x[N+1]** 5 + 726.38 * x[N+1] **4 - 855.63 * x[N+1] ** 3 + 490.17 * x[N+1] ** 2 - 159.61 * x[N+1] + 99.509;

Temp[N]=Temp[N+1];

hx[N+1]=-0.0167 * Temp[N+1]** 3 + 4.0511 * Temp[N+1]** 2 - 293.95 * Temp[N+1] + 8850.2;

hxc = hx[N + 1] - (((xc * Cpm) + ((1 - xc) * Cpw)) * ((Temp[N + 1]*1.8+32)-(Temp[N] * 1.8 + 32)));

hx[N] = hx[N + 1] - (((xdn * Cpm) + ((1 - xdn) * Cpw)) * ((Temp[N + 1]*1.8+32)-(Temp[N] * 1.8 + 32)));}

if (N==Nstg-1){

L[N]=0;

xb=xbn;

x[N]=xbn;

y[N]=11.21 * xbn ** 5 - 33.47 * xbn ** 4 + 37.88 * xbn** 3 - 20.28 * xbn** 2 + 5.65 * xbn; //Vapor in equilibrium with reboiler composition

if (V[N]<0 || y[N]<0 || y[N]>1){

constraint=constraint+1;

break;}

if (x[N]<0 || x[N]>x[N-1]){

constraint=constraint+1;

break;}
```

```
Temp[N] =-236.53 * x[N]** 5 + 726.38 * x[N] **4 - 855.63 * x[N] ** 3 + 490.17 * x[N] ** 2 -
159.61 * x[N] + 99.509;

hx[N]=-0.0167 * Temp[N]** 3 + 4.0511 * Temp[N]** 2 - 293.95 * Temp[N] + 8850.2;

Hy[N]=91.792*Temp[N] + 11335;

}

if (N!=0 && N!=Nstg-1 ){

if (N==1){

V[N]=PdtD[0]+L[N-1];}

else{

V[N]= V[N-1]-L[N-2]+L[N-1]-F[N-1];

if (V[N]>0){

y[N]=((V[N-1]*y[N-1]-L[N-2]*x[N-2]+L[N-1]*x[N-1]-F[N-1]*Z)/V[N]);}}

if (V[N]<0|| y[N]<0|| y[N]>1){

constraint=constraint+1;

break;}

if (N!=1){

if(y[N]>y[N-1]){

constraint=constraint+1;

break;}

x[N]=-5.755 * y[N] **5 + 12.989 * y[N]** 4 - 8.682 * y[N] ** 3 + 2.532 * y[N]** 2 - 0.079 *
y[N];

if (x[N]<0 || x[N]>x[N-1]){

constraint=constraint+1;

break;}}

Temp[N] =-236.53 * x[N]** 5 + 726.38 * x[N] **4 - 855.63 * x[N] ** 3 + 490.17 * x[N] ** 2 -
159.61 * x[N] + 99.509;

hx[N]=-0.0167 * Temp[N]** 3 + 4.0511 * Temp[N]** 2 - 293.95 * Temp[N] + 8850.2;

Hy[N]=91.792*Temp[N] + 11335;}}
```

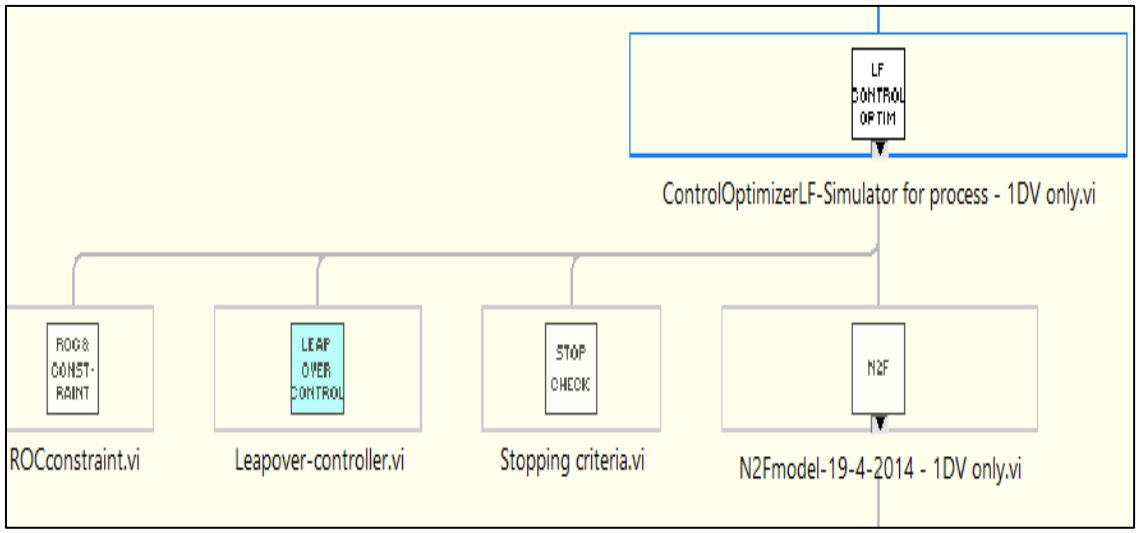
```
Qc=V[1]*(Hy[1]-hxc);

for (N=0; N<=Nstg-2; N++){

if (constraint>0){

break;}

if(N==0){

dev[N] = (F[Feedstg]*Z-PdtD[0]*xd-PdtB[Nstg-1]*xb)**2;}

else{

dev[N] = (((V[N] * Hy[N] + L[N] * hx[N]  - F[N] * hf - L[N - 1] * hx[N - 1] - V[N + 1] * Hy[N
+ 1])) ** 2)/(eceb**2);}

totdev[N] = dev[N];

OFnew= sumdev1 + totdev[N];

sumdev1 = OFnew;}
```

APPENDIX B

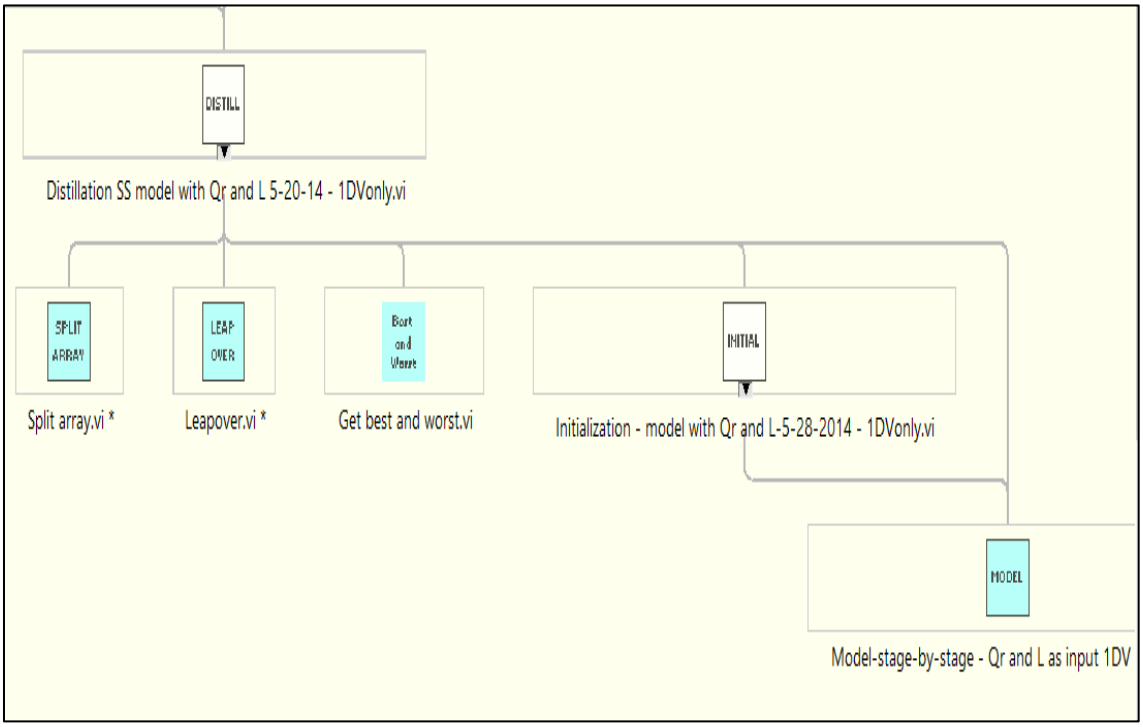
Hierarchy of Distillation MPC Simulator in LabVIEW



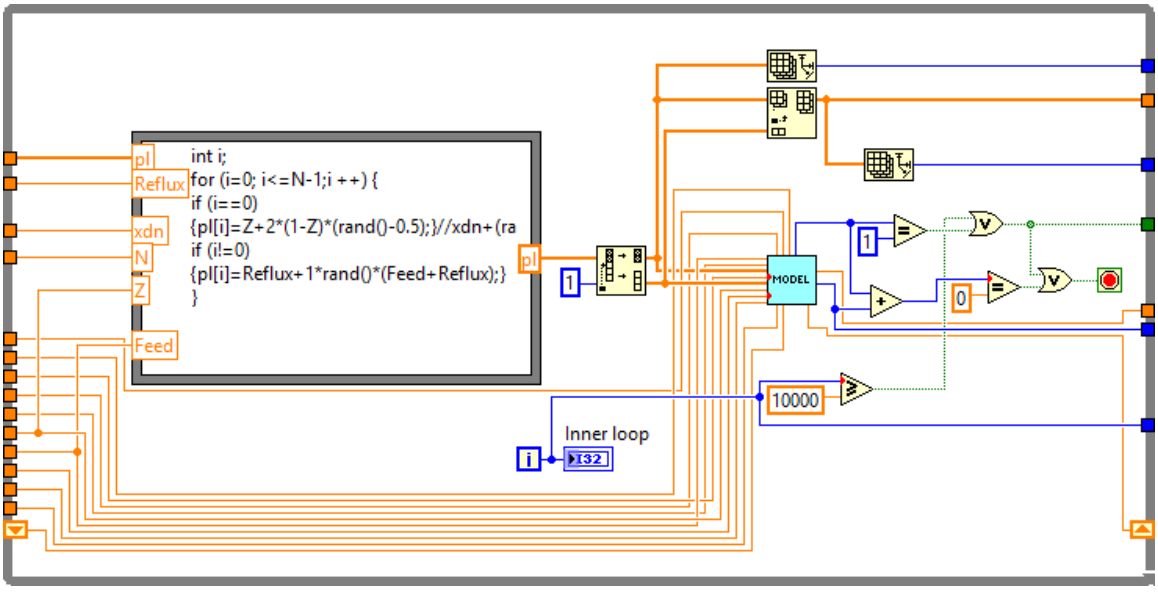
Hierarchy of Controller Optimizer



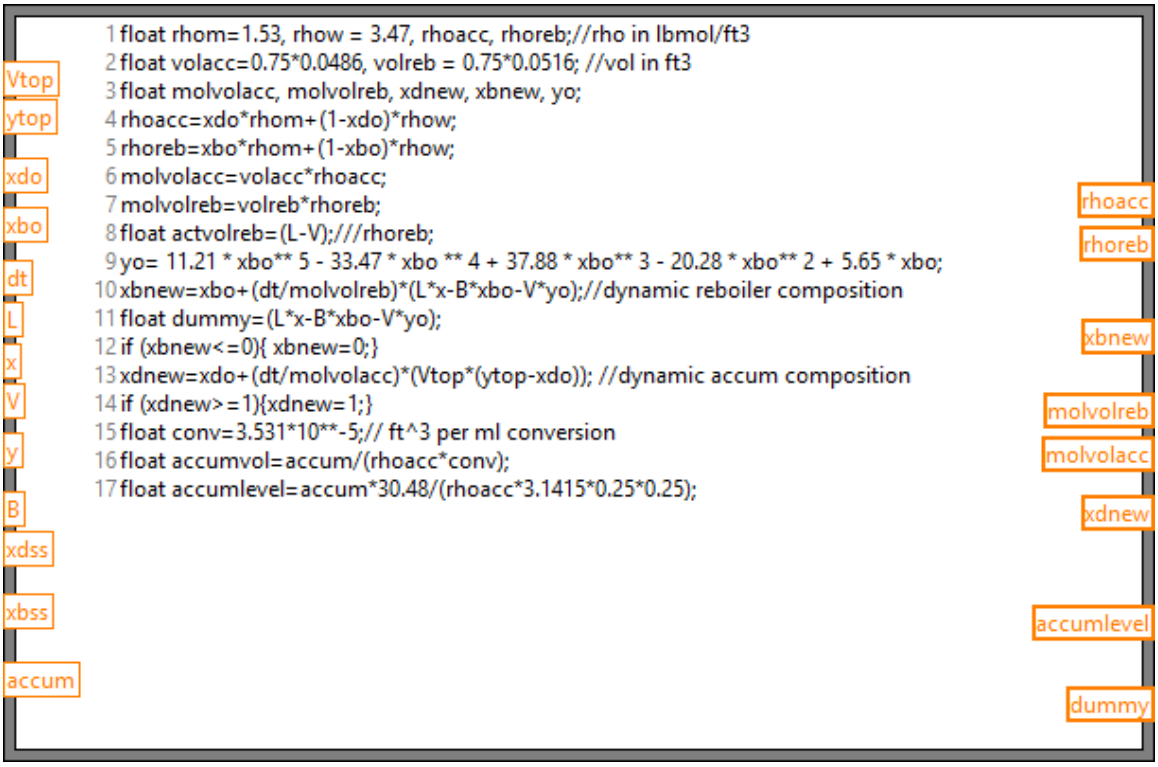
Hierarchy of Steady State Distillation Model



Screenshot of Steady State Distillation Initialization

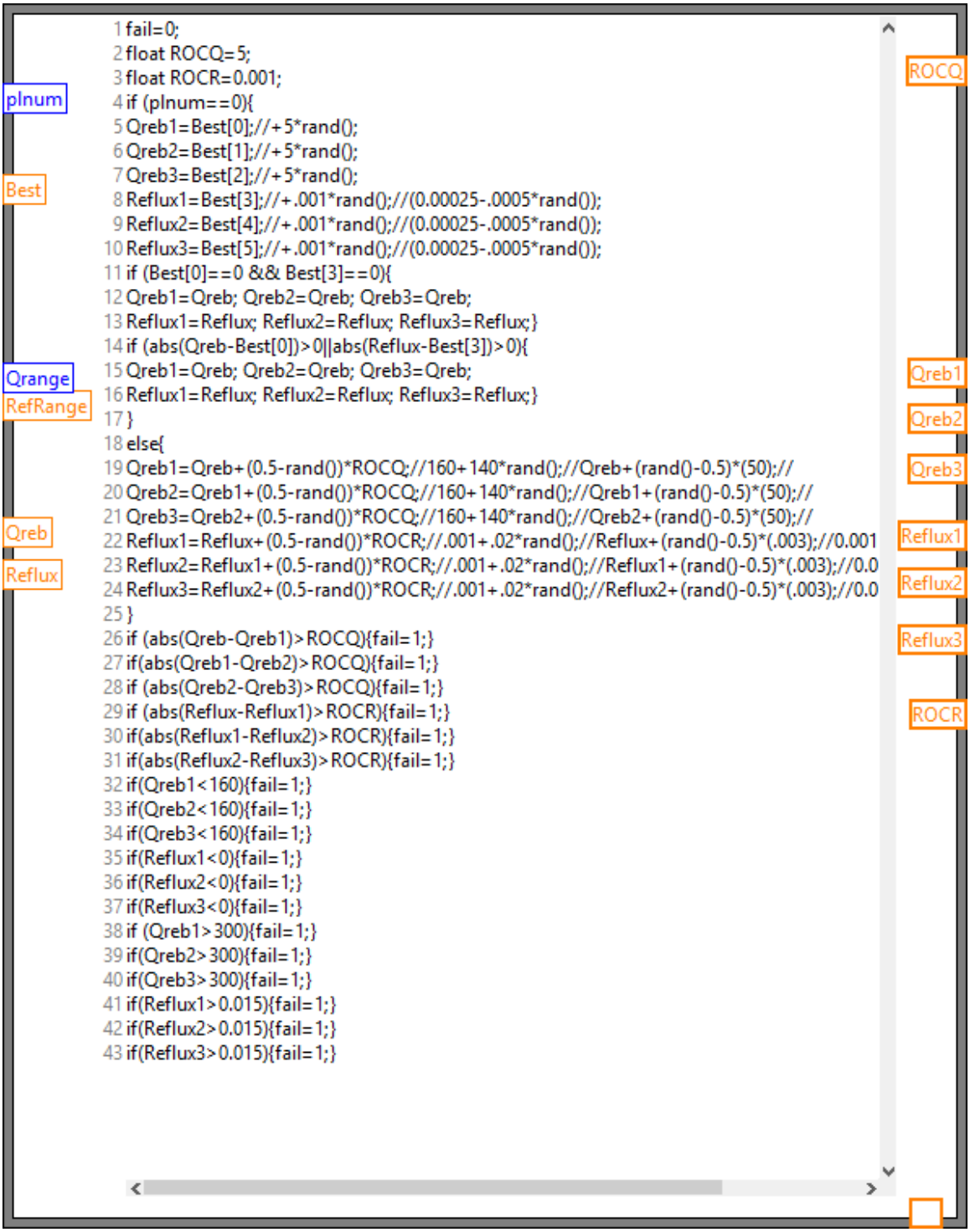


Screenshot of P2N Model

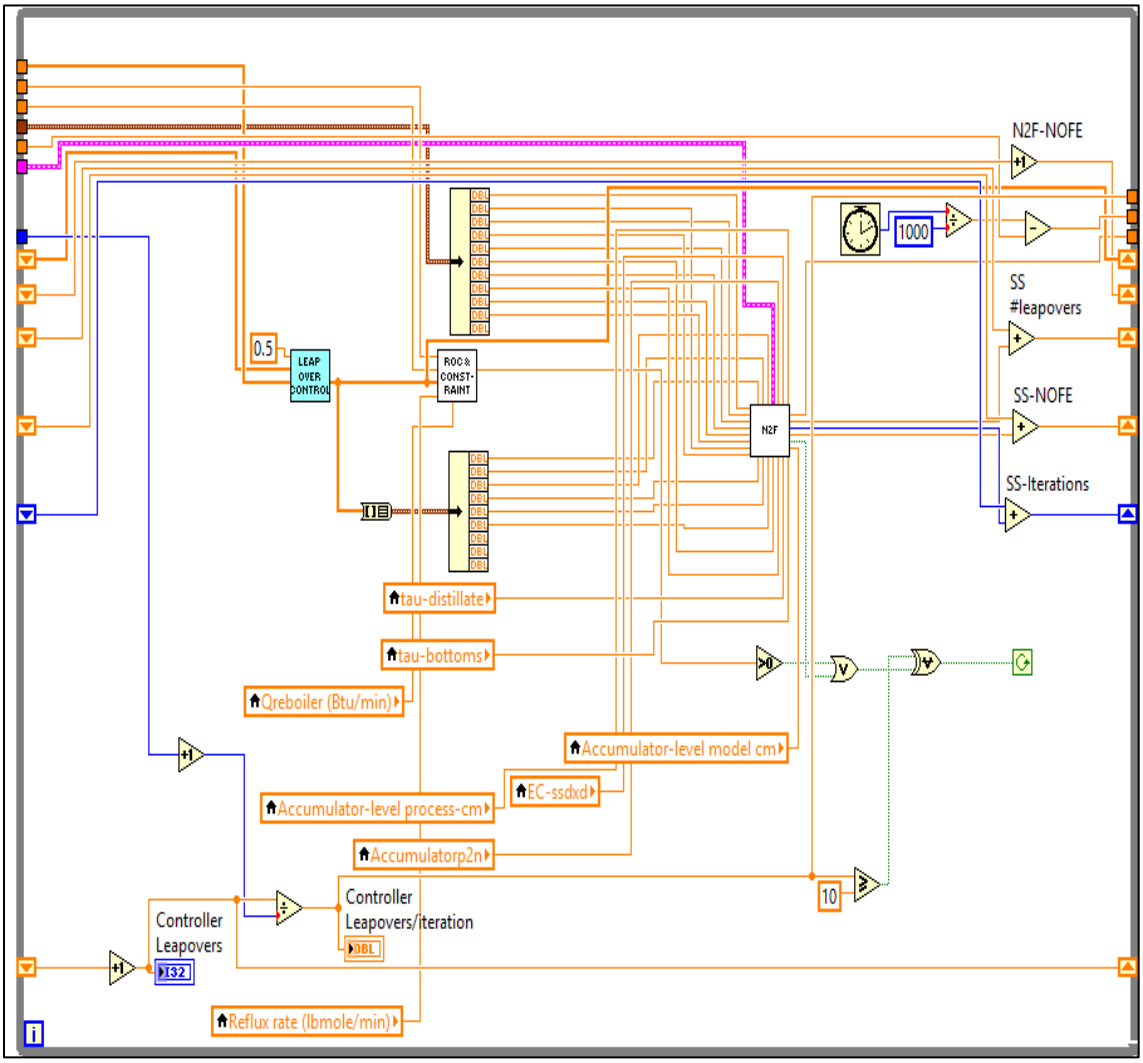


Screenshot of N2F Model

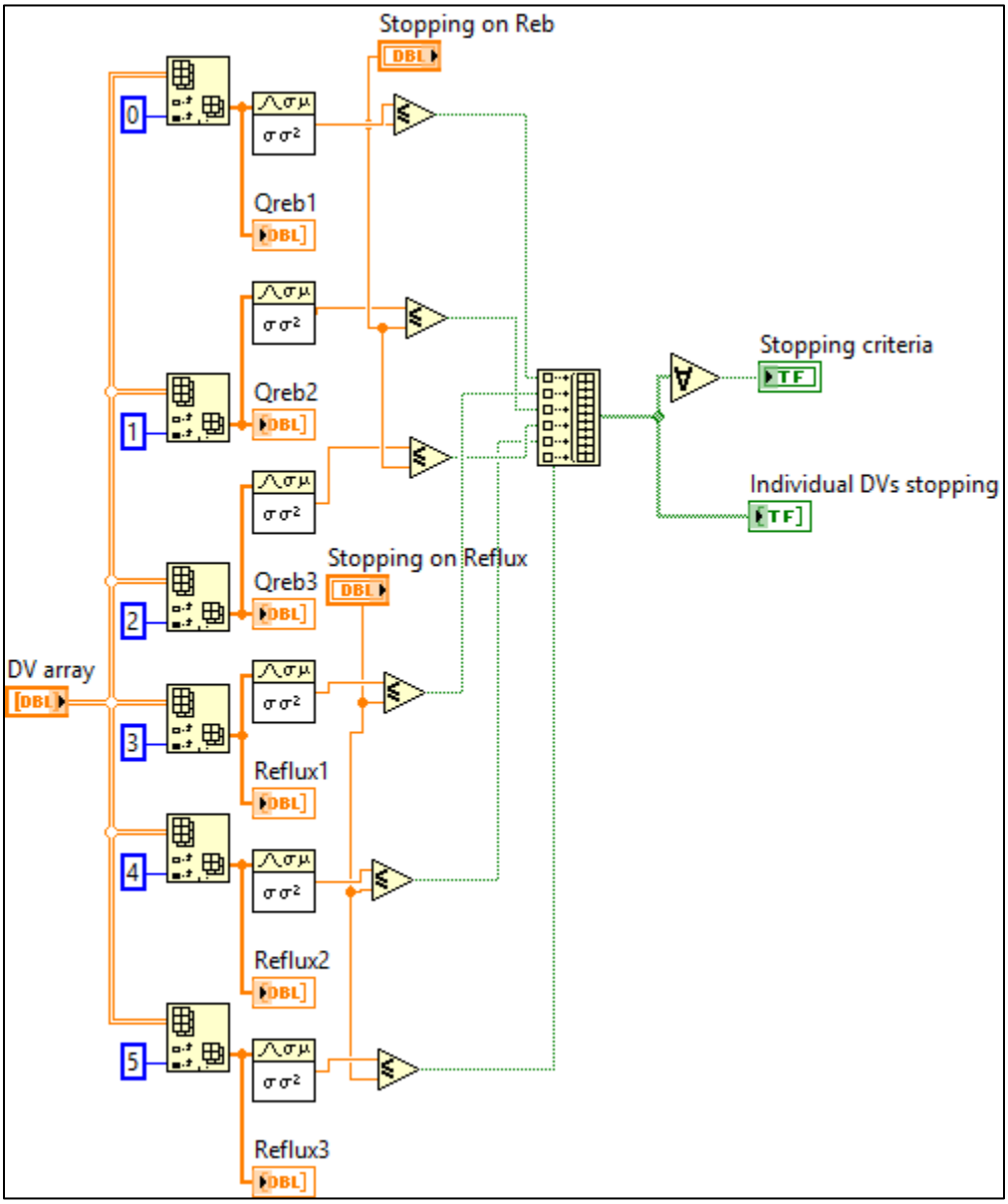
Screenshot of Controller Initialization



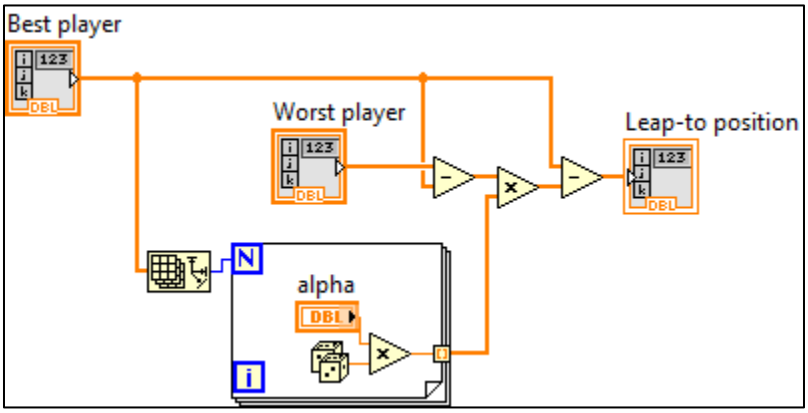
Screenshot of Controller Leapover



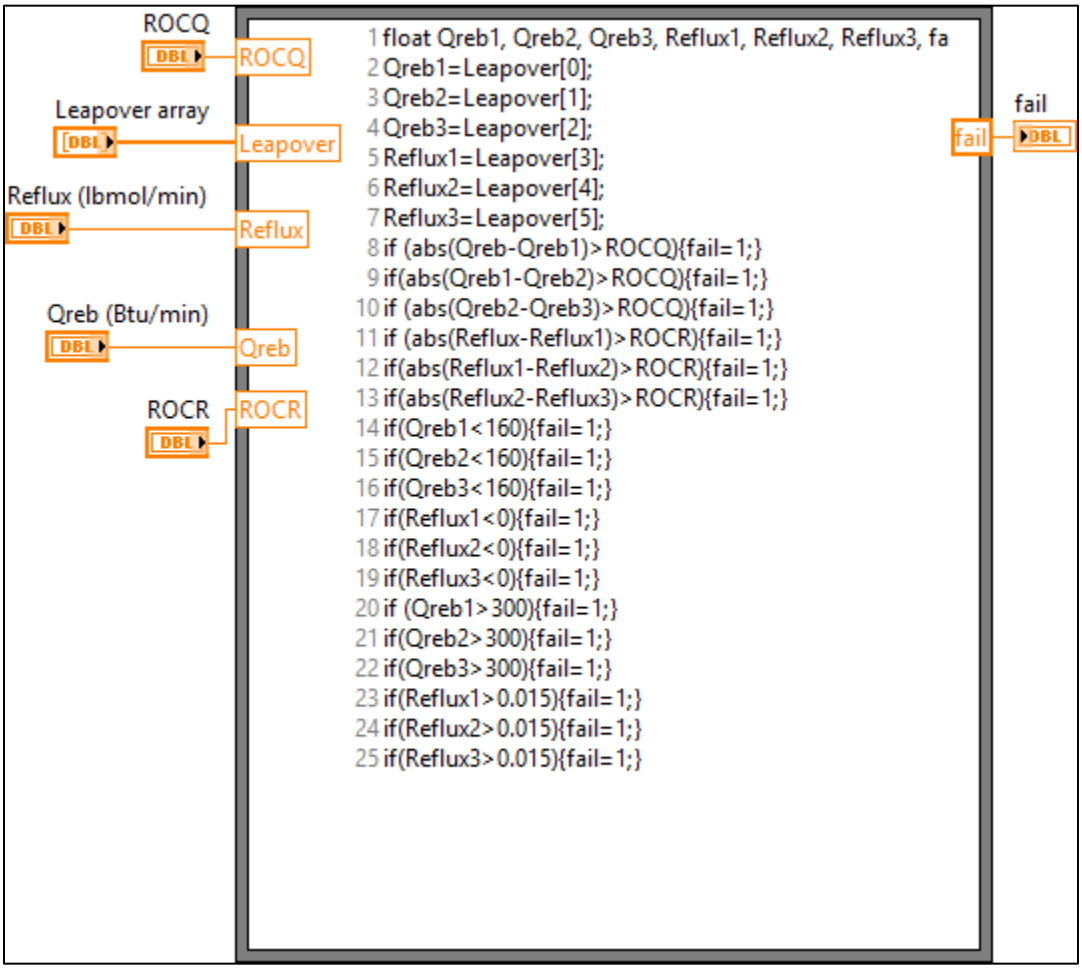
Screenshot of Controller Stopping Crieteria



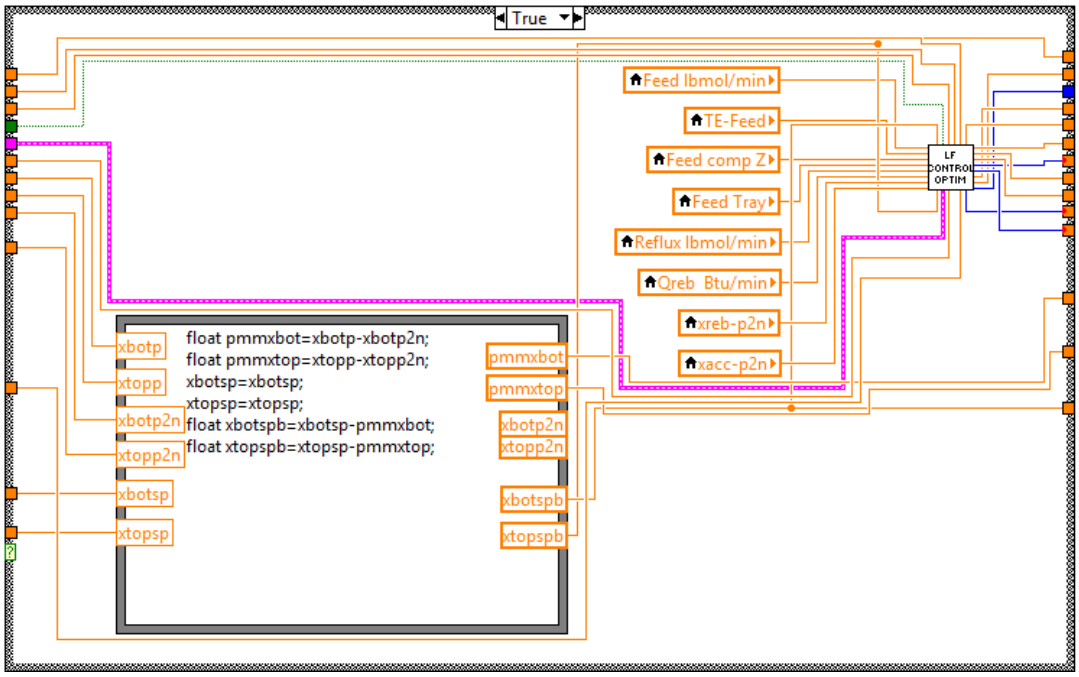
Screenshot of a Leapover



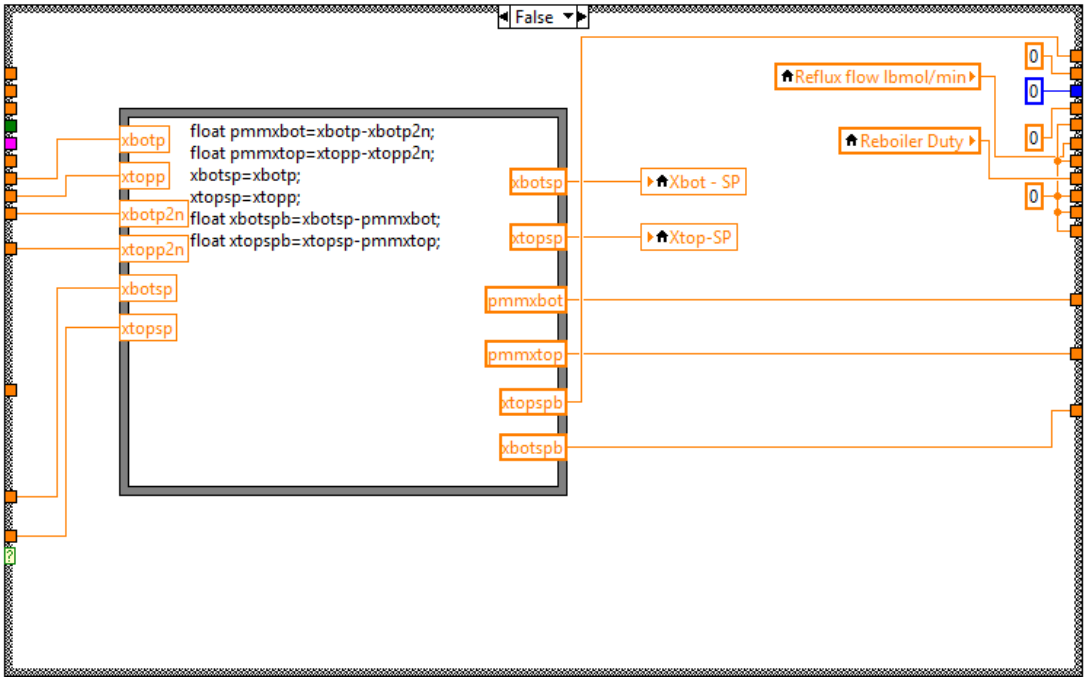
Screenshot of Rate of Change Constraint



Screenshot of Manual Mode



Screenshot of Auto Mode



VITA

Type Anand Govindarajan

Candidate for the Degree of

Doctor of Philosophy

Thesis: ACCELERATING CONVERGENCE OF LEAPFROGGING OPTIMIZATION – APPLICATIONS TO NONLINEAR PROCESS MODELING AND NONLINEAR MODEL PREDICTIVE CONTROL

Major Field: Chemical Engineering

Biographical:

Education:

Completed the requirements for the Doctor of Philosophy in Chemical Engineering at Oklahoma State University, Stillwater, Oklahoma in July, 2014.

Completed the requirements for the Master of Science in Chemical Engineering at Oklahoma State University, Stillwater, Oklahoma in 2011.

Completed the requirements for the Bachelor of Technology in Chemical Engineering at SSN College of Engineering (Anna University), Chennai, Tamil Nadu, India in 2008.

Experience:

Teaching Associate – Unit Operations and Advanced Control Lab 2012-2014
Engineering Intern – Fractionation Research Inc 5/2013 – 8/2013
Technical Intern – Sun to Market Solutions Inc 6/2010 – 8/2010
Research Associate – SSN Research Center 8/2008 – 7/2009
Graduate Engineer (Trainee) – VA Tech WABAG India 6/2008 – 10/ 2008

Professional Memberships:

International Society of Automation
OSU Automation Society
Chemical Engineering Graduate Student Association
Omega Chi Epsilon