INTEGRATION OF PROCESS DESIGN, SCHEDULING, AND CONTROL VIA MODEL BASED MULTIPARAMETRIC PROGRAMMING

A Thesis

by

BARIS BURNAK

Submitted to the Office of Graduate and Professional Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

Chair of Committee,	Efstratios N. Pistikopoulos
Committee Members,	Mahmoud El-Halwagi
	Costas Kravaris
	Le Xie
Head of Department,	Arul Jayaraman

August 2020

Major Subject: Chemical Engineering

ABSTRACT

The conventional approach to assess the multiscale operational activities sequentially often leads to suboptimal solutions and even interruptions in the manufacturing process due to the inherent differences in the objectives of the individual constituent problems. In this work, integration of the traditionally isolated process design, scheduling, and control problems is investigated by introducing a multiparametric programming based framework, where all decision layers are based on a single high fidelity model. The overall problem is dissected into two constituent parts, namely (i) design and control, and (ii) scheduling and control problems. The proposed framework was first assessed on these constituent subproblems, followed by the implementation on the overall problem. The fundamental steps of the framework consists of (i) developing design dependent offline control and scheduling strategies, and (ii) exact implementation of these offline rolling horizon strategies in a mixed-integer dynamic optimization problem for the optimal design. The design dependence of the offline operational strategies allows for the integrated problem to consider the design, scheduling, and control problems simultaneously. The proposed framework is showcased on (i) a binary distillation column for the separation of toluene and benzene, (ii) a system of two continuous stirred tank reactor, (iii) a small residential heat and power network, and (iv) two batch reactor systems. Furthermore, a novel algorithm for large scale multiparametric programming problems is proposed to solve the classes of problems frequently encountered as a result of the integration of rolling horizon strategies.

DEDICATION

To Kadriye and Oya Burnak.

ACKNOWLEDGMENTS

Before the reader starts exploring the fascinating facets of my five years of labor, I would like to spare a few lines for those who have supported and inspired me throughout this adventure as a small gesture to express my gratitude.

I would like to thank my advisor, Professor Efstratios N. Pistikopoulos, for his help and supervision during my graduate life at Texas A&M. I also thank my thesis committee members, Professor Mahmoud El-Halwagi, Professor Costas Kravaris, and Professor Le Xie for their constructive criticism to push the limits of the potential of my study.

As one of the perquisites of being a part of the Pistikopoulos research group, I am fortunate enough to have collaborated with some of the greatest minds and talents in their fields. As my mentor and guide in my early years, Nikolaos A. Diangelakis provided me with the warmest of welcomes in the group and the highest of standards in research. Justin Katz had been a great ally to have to share the sometimes overwhelming burden. My thanks to Melis Onel, who has the gift to make the collaboration much easier than it should be with her enthusiasm. Albeit in my last year, I had the privilege to work with William Tso and I profoundly enjoyed every bit of our exquisite discussions. I also thank Professor Maria M. Papathanasiou, Styliani Avraamidou, Yuhe Tian, Iosif Pappas for making these potentially exhausting collaborations seamless and smooth.

I am most grateful to Salih Emre Demirel, a true companion, for standing by my side when the times were tough.

I thank the current and former members of our group, Ioana Nascu, Richard Oberdieck, Maria Papathanasiou, Onur Onel, Gerald Ogumerem, Burcu Beykal, Pankaj Goel, Doga Demirhan, Denis Su-Feher, Yuhe Tian, Chris Gordon, Stefanos Baratsas, Cory Allen, Shayan Niknezhad, Rahul Kakodkar, Dustin Kenefake, Shivam Vedant, Vaishnav Meduri, Naseem Masoud, Marcello Di Martino, and Hari Ganesh. As an honorary member of our group, I also thank Aurora Vargas for being the best neighbor for two years!

My anchors to my homeland, my friends, deserve a moment of appreciation for their sup-

port. Eray Temur and Derya Dokur Temur, Ayşe Nur Çelik, Görkem Özdemir, Umur Atam and Cansu Çelebi Atam, Melodi Baç, Elifnaz Önder, Meriç Neşeli, Doruk Neşeli, Fuat Sakirler, Tuğhan Caner, Gizem Çelikboya, Merve Say Keleş and İlteber Kaan Keleş, Alper Benli, Serdar Özsezen, Bilge Kerem Aksakal, and İrem Topal Kement and Erdi Kement. They are the unsung heroes, helping me to keep my sanity from thousands of miles away. My former lab mate Burcu Karagoz had been a great host for me in Pittsburgh, even a greater guest every time she had visited Austin.

I appreciate all the good times I had with my fellow colleagues from ERB and the delightful chess nights, so I thank Spyros Tsolas, Abhinav Narasingam, Akhil Arora, Ahmed Harhara, Niranjan Sitapure, and Mohammed Bangi.

Needless to say my beloved family, Oya and Kadriye Burnak, have supported me unconditionally from day one. Realizing how rare this can be, I deeply value their encouragement to keep going. Lastly, I would like to thank my father, Metin Burnak, whose footsteps paved the way for me to become an engineer, but much before that, to be a good person.

To end this section on a high note, I would like to thank the following names for their direct or indirect contributions to this thesis in no particular order: Jack Skellington, Kim Wexler, Ramona Flowers, Iroh, Marty McFly, Kobe Bryant, Stephen Curry, Helena Bonham Carter, Max, Søren Kierkegaard, Margaery Tyrell, Quentin Tarantino, Christoph Waltz, Thor, Erik, Woody Allen, Emma Stone, İhsan Oktay Anar, Sevgi Soysal, Alan Parsons, Wes Anderson, Andrew Latimer, Carolina Eyck, Lindsay Ellis, Hugh Laurie, Hümeyra, Ferhan Şensoy, Dilara Sakpınar, immanuel tolstoyevski, Albert Camus, Jean-Paul Sartre, Sweeney Todd, Albus Percival Wulfric Brian Dumbledore, Qui-Gon Jinn, Demet Evgar, Mufasa, Michael Scott, HAL 9000, Groot, Ivan Fyodorovich Karamazov, Olric, Rick Sanchez, Pierre-Joseph Proudhon, Karl Marx, Morgan James, Jean Valjean, Bruce Dickinson, Mike Portnoy, Walter H. White, Benjamin Linus, Edgar Allan Poe, Sven Magnus Øen Carlsen, Igor Stravinsky, Fırat Budacı, Ersin Karabulut, Austin Wintory, and Diane Nguyen. Thank you all for being a part of my life.

CONTRIBUTORS AND FUNDING SOURCES

Contributors

This work was supported by my advisor Professor Efstratios N. Pistikopoulos, and a thesis committee consisting of Professor Mahmoud El-Halwagi and Professor Costas Kravaris of the Department of Chemical Enginnering, and Professor Le Xie of the Department of Electrical and Computer Engineering.

The methodology and the implementation of the work in Chapters 2, 3, and 4 were completed in collaboration with Dr. Nikolaos A. Diangelakis from Imperial College London and Dr. Justin Katz from the Department of Chemical Engineering. Dr. Katz and I also developed the algorithm discussed in Chapter 6 together. The work presented as Supplementary Material in Chapter 7 was completed in collaboration with Dr. Melis Onel from the Department of Chemical Engineering. All other work conducted for the thesis was completed by the student independently. A portion of the work presented in this thesis was published in various journals, indicated in their respective chapters.

Funding Sources

I acknowledge the financial support from the NSF (Grant no. 1705423) under the project titled "SusChEM: An integrated framework for process design, control and scheduling [PAROC]", Texas A&M Energy Institute, Shell Oil Company, Rapid Advancement in Process Intensification Deployment (RAPID) Institute, Clean Energy Smart Manufacturing Innovation Institute (CESMII), and U.S. National Institute of Health (NIH) grant P42 ES027704. Portions of this research were conducted with the advanced computing resources provided by Texas A & M High Performance Research Computing. The content of this thesis does not necessarily represent the official views of these institutes.

TABLE OF CONTENTS

Page

LIST OF FIGURES

FIGURE Page

LIST OF TABLES

TABLE Page

1. INTRODUCTION AND OVERVIEW[1]

The complexity of decision making problems in the process industry has conventionally resulted in isolation of decisions with respect to the time scales of their effects on the operation, ranging from years-spanning process synthesis and design to seconds-long process control decisions. Apart from the discrepancies in the time scales, the conventional decision layers differ in their objectives. The long term operating decisions, such as the planning problem, aim to maximize the profit, while the scheduling problem addresses the feasibility of the operation by allocating the raw materials, utilities, and the process equipment in a shorter horizon. The dynamic performance of the scheduled operation is assessed and maintained a posteriori by the process control decisions, where the time steps are orders of magnitude smaller than the scheduling decisions [1]. In addition to the operational decisions, the "here-and-now" process design problem is another long term decision that has significant impact on the future operating strategies and hence, the profitability, feasibility, and the performance of the operation [2]. This established hierarchy of decision layers largely neglects their interactions and suggests the information to flow dominantly in descending order in the time scales the layers span [1].

However, overlooking the bidirectional relationship between the decision layers may lead to infeasible operations, forcing the process decisions to be overly conservative. The efforts to maintain feasible operations without considering all the constituent layers in the decision hierarchy simultaneously results into deviation from the optimal decisions. Process Systems Engineering (PSE) community has been accumulating formidable knowledge and know-how on mathematical modeling techniques to improve our understanding on process design and operations, and developed efficient tools to solve these advanced models [3]. It has been long established that the early design problem should be studied simultaneously with the operational time-variant decisions to improve the operability and flexibility of the process under variable internal and external plant conditions,

[1]Portions of this chapter have been adapted from Burnak, B., Diangelakis, N.A., Pistikopoulos, E.N., Towards the Grand Unification of Process Design, Scheduling, and Control – Utopia or Reality?, Processes 2019, 7, (7), 461.

and consequently to achieve more reliable, economically more favorable, and inherently safer processes. The most recent efforts towards simultaneous consideration of design and operational decisions explore effective methodologies to integrate the short term process regulatory decisions (process control) and longer term economical decisions (scheduling) through mixed-integer dynamic optimization (MIDO) formulations. The proposed solution tools and techniques for this class of integrated problems include (i) discretizing the dynamic high fidelity representation of the process through orthogonal collocation on finite elements followed by solving a mixed-integer nonlinear programming problem [4], (ii) "back-off" approach to ensure constraint satisfaction under some assumed worst-case scenario [5, 6, 7], and (ii) multiparametric programming to explicitly represent the operational strategies to derive tractable and equivalent MIDO formulations [8].

In this introduction chapter, we present a historical perspective on the development and progress of modern process design techniques that account for the dynamic variability introduced by the process control and scheduling decisions. In retrospect, we observe the evolution of methodologies from fundamental analyses on design and process uncertainty at steady state to dynamic complex models that explicitly encapsulate the scheduling and control decisions, as illustrated in Figure 1.1, and summarized as follows.

i. *Flexibility analysis and flexibility index.* The early stages for design optimization under uncertainty. The studies here analyze the steady state feasibility of a nominal process design under a set of unknown process parameters and unrealized operating decisions, as we will discuss in Section 1.1.

ii. *Dynamic resilience and controllability analysis.* Here, the researchers investigate the dynamic response of a system in closed loop, its interdependence with process design, and attempt to develop the "perfect controller" simultaneously with the process that the controller can act on. Such attempts will be demonstrated in Section 1.2.

iii. *Complete integration of design, control, and operational policies.* The focus of the most recent studies in the field. The goal is to model tractable dynamic design optimization prob-

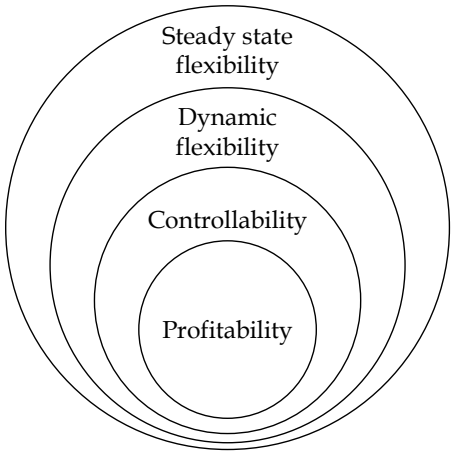


Figure 1.1: A Venn diagram representation of major operability indices and their relation with process economics. It is interesting to note that the design optimization approaches started from the outermost layer, and with the advance of modeling techniques, they have progressed towards the center for guaranteed operability, which delivers the optimal process economics (Reprinted with permission from [9]).

lems that account for the scheduling and control decisions to guarantee the operability and even profitability of the operation under all foreseeable conditions. These approaches will be discussed in Section 1.3.

Clearly, it would be inaccurate and redundant trying to reduce down the individual research efforts to a single category. The literature is noticeably diverse in this field with numerous different approaches. However, we find it useful to classify into certain schools of thought that are also in alignment with the historical progress of the field. In Section 1.4, we further seek to pose the pivotal questions on future challenges and opportunities for the seamless integration of the design, scheduling, and control problems based on the cumulative knowledge of the PSE community and the current trends in the academia.

1.1 Early efforts in design optimization under uncertainty

The ongoing collective efforts towards the grand unification of design, scheduling, and control was inaugurated through steady state design under uncertainty in plant conditions. Takamatsu et al. (1970) [10] estimated the undesirable effects of variations in system parameters, measured

process disturbances, and manipulated variables on plant performance by sensitivity analysis on a linearized model. Nishida et al. (1974) [11] adopted the notion of sensitivity analysis to structure a min-max problem for design optimization, presented by Eq. 1.1.

$$\begin{aligned} \min_{des} \max_{\theta} \quad & C(x, des, \theta) \\ s.t. \quad & h(x, des, \theta) = 0 \\ & g(x, des, \theta) \leq 0 \\ & \underline{\theta} \leq \theta \leq \overline{\theta} \end{aligned} \tag{1.1}$$

where x is the vector of states of the system, des is the vector of design variables including the steady state manipulated variables, θ is the vector of parameters that agglomerates the system uncertainties and process disturbances. Equation 1.1 is one of the first notable attempts to systematically assess the trade-off between minimizing the investment cost and improving the flexibility of the process design. However, this strategy yields conservative solutions since it does not distinguish the time-invariant design variables and time-variant manipulated variables. Grossmann and Sargent (1978) [12] remedied this issue by treating the time-sensitive variables (i.e. manipulated actions and design variables that can be modified in the future) and fixed design variables separately. They further adopted the parametric optimal design problem proposed by Kwak and Haug (1976) [13], and formulated an objective function to minimize the average cost over the expected range of parametric uncertainty, as presented by Eq. 1.2.

$$\begin{aligned} \min_{u,des} \quad & E\{C(x, u, des, \theta)\} \\ s.t. \quad & \max_{\theta \in \Theta} \quad g_i(x, u, des, \theta) \leq 0, \quad i = 1, 2, ..., t \end{aligned} \tag{1.2}$$

where the expected cost function is defined the joint probability distribution of the parameter set θ. Equation 1.2 requires solving infinite nonlinear programming (NLP) problems. Grossmann and Sargent (1978) [12] proposed an efficient solution procedure for a special case of Eq. 1.2, where each constraint g_i is monotonic in θ, through discretization of the problem over the parameter

space. However, solving the NLP problem at a finite number of θ realizations does not ensure the feasibility of the design. This issue is addressed by Halemane and Grossmann (1983) [14] through reformulating an equivalent design feasibility constraint as presented by Eq. 1.3.

$$\max_{\theta \in \Theta} \min_{u \in U} \max_{i \in I} g_i(x, u, des, \theta) \leq 0 \tag{1.3}$$

The max-min-max problem in Eq. 1.3 mathematically expresses the feasibility question "For all the uncertainty realizations Θ, does there exist a control action u such that the constraint set g is feasible?". Equation 1.3 was employed in a multiperiod design optimization problem, where the deterministic uncertain parameter θ was allowed to vary within a prespecified range [14]. The feasibility constraint then laid the foundation for the concept of feasibility index, F, proposed by Swaney and Grossmann (1985) [15], as given by Eq. 1.4.

$$\begin{aligned} F = \max \quad & \delta \\ s.t. \quad & \max_{\theta \in \Theta} \min_{u \in U} \max_{i \in I} g_i(x, u, des, \theta) \leq 0 \\ & T(\delta) = \{\theta \mid (\theta^{nom} - \delta \Delta \theta^{-}) \leq \theta \leq \theta \mid (\theta^{nom} + \delta \Delta \theta^{+}\} \end{aligned} \tag{1.4}$$

where T is the hyperrectangle for the uncertain parameters, δ is the scaled parameter deviation, and the superscript nom denotes nominal conditions. Equation 1.4 is the first significant attempt to quantify the degree of flexibility of a process design, and has been exploited by numerous studies on design optimization and process operability. However, Eq. 1.4 constitutes a nondifferentiable global optimization problem and is still quite challenging to solve. Therefore, it requires simplifying assumptions and approximations to maintain a tractable problem. Swaney and Grossmann (1985) [16] introduced a heuristic vertex search method and an implicit enumeration scheme for the special case where the critical uncertainty realizations are assumed to lie at the vertices of the hyperrectangle $T(\delta)$. Clearly, this assumption fails to hold when the feasible space of the design problem is non-convex. Grossmann and Floudas (1987) [17] relaxed this assumption by developing a mixed-integer nonlinear programming (MINLP) problem for the feasibility test presented

by Eq. 1.3. They further proposed an active constraint strategy for the solution of the resulting MINLP. The mixed-integer formulation also provides a systematic approach to consider all possible critical uncertainty realizations without exhaustive enumeration. The proposed formulation was utilized for synthesis of a heat exchanger network with uncertain stream flow rates and temperatures [18]. The case of linear constraints reduces to an MILP problem, for which global solution is attainable by standard branch and bound enumeration techniques [17, 19, 20]. Bansal et al. (2000) [21] developed a computationally efficient theory and algorithm based on multiparametric programming techniques for this special case of flexibility analysis problems. The authors derived explicit expressions for the flexibility index as explicit functions of the continuous design variables. Pistikopoulos and Grossmann (1988a, 1988b, 1988c) used the flexibility test with linear constraints for optimal retrofit design [22, 23, 24, 25] and redesign under infeasible nominal uncertainties [26]. Although these approaches are effective and promising to handle the design uncertainty, they require solving nested optimization problems, which poses a major challenge to solve complex and large scale problems in a reasonable time. Raspanti et al. (2000) [27] proposed replacing the complementarity conditions of the lower level optimization problems with a well-behaved, smoothed nonlinear equality constraints, namely Kreisselmeier and Steinhauser function [28] and Chen and Mangasarian smoothing function [29].

One of the common assumptions in these approaches is the known bounds of the uncertainties, which is rarely the case in real world industrial applications. Pistikopoulos and Mazzuchi (1990) [30] and Straub and Grossmann (1990, 1993) [31, 32] extended the flexibility test by assuming a probability distribution model for the parameter uncertainty, which improved the economical performance of the design optimization problem by addressing the "conservativeness" of the solution.

Another common assumption of these approaches is the steady state operation of the plant design, which creates a significant limitation on the applicability of the methodologies. Although steady state assumption holds true for the dominant life cycle of the plant operation, design optimization problem may fail to ensure the operability under transient behaviors such as startup or shutdown and transitions between different operating conditions. Dimitriadis and Pistikopoulos

(1995) [33] proposed a dynamic feasibility index for the systems that are described by differential algebraic equations (DAE) subject to time-varying constraints. However, the time-dependent uncertainty in their formulation dictates to solve infinitely many dynamic optimization problems. Therefore, the authors assumed that the critical scenarios of uncertainties are known and lie on the vertices of the time-varying uncertainty space, similar to Swaney and Grossmann (1985) [15]. The simplifying assumption reduced the problem to the form given by Eq. 1.5.

$$
\begin{aligned}
DF(des) = \max_{\delta, u(t), t} \quad & \delta \\
s.t. \quad & \dot{x} = f(x(t), u(t), des, \theta(t), t), \quad x(0) = x_0 \\
& g(x(t), u(t), des, \theta(t), t) \leq 0 \\
& \theta(t) = \theta^N(t) + \delta \Delta\theta^c(t) \\
& \delta \geq 0, \quad \underline{u}(t) \leq u(t) \leq \overline{u}(t)
\end{aligned}
\tag{1.5}
$$

where the time dependence of the variables constitute a dynamic optimization problem, and the solution was determined by control vector parameterization techniques [33]. Dynamic flexibility has been widely utilized in numerous design optimization applications including batch processes [34], separation systems [35, 36, 37, 38, 39], reaction systems [40], and heat exchanger network synthesis [41, 42, 43].

The dynamic assessment of the plant feasibility under uncertainty has been also studied through exploiting the multiperiod design optimization formulation proposed by Halemane and Grossmann (1983) [14]. Varvarezos et al. (1992) [44] implemented an outer-approximation approach to solve the multiperiod multiproduct batch plant problems operating with single product campaigns, which was formulated as an MINLP. Pistikopoulos and Ierapetritou (1995) [45] considered stochastic process uncertainty and proposed a two-stage decomposition that can handle convex nonlinear problems.

As presented in this section, the early studies on integrated design optimization have primarily focused on (i) investigating the range of operation (flexibility) of a nominal design configuration under foreseeable conditions, and (ii) determining the "best" possible trade-off between the invest-

ment cost and the capability of handling variations in the internal and external operating conditions. These studies mostly considered open loop processes, under the traditional assumption that controller design is a sequential task to process design. However, most processes in industry are operated in closed loop, and the controller schemes inherently alter the process dynamics, rendering the open loop flexibility analyses of lesser relevance. In other words, an "attainable" operating point according to open loop flexibility analysis may actually be an infeasible point in closed loop. Realizing the shortcomings of open loop flexibility analyses, researchers began investigating the "controllability" of process systems, and the interdependence of process control and design decisions. In the following section, we present a retrospective background on the integration of process control in the design optimization problem.

1.2 Integration of process control in design optimization

The initial efforts towards the integration of process control and design problems established a fundamental understanding on the interdependence of the two decision making mechanisms. The most pronounced school of thought in the early years to evaluate the controllability of the process design is "dynamic resilience", as conceptually defined by Morari (1983a, 1983b) [46, 47].

Morari (1983) [46] described dynamic resilience as "the ability of the plant to move fast and smoothly from one operating condition to another and to deal effectively with disturbances". This depiction implies that there is not a clear cut distinction between flexibility, which was discussed in Section 1.1, and resilience. However, Grossmann and Morari (1983) [48] pointed out the main difference as "resiliency refers to the maintenance of satisfactory performance despite adverse conditions while flexibility is the ability to handle alternate (desirable) operating conditions". This distinction is the primary motive for the majority of the flexibility analyses to study steady state operations, while the resilience deals with the dynamic operations, as we will discuss in this section.

Dynamic resilience, as described by Morari (1983) [46], aims to find the "perfect controller" that is allowed by the physical limitations of the system to assess the controllability of the process by using the internal model control (IMC) structure. The proposed technique decomposes the

system transfer function $\tilde{G}$ into (i) a non-singular matrix $\tilde{G}_-$ to design the perfect controller $\tilde{G}_-^{-1}$, and (ii) a singular matrix $\tilde{G}_+$ to generate dynamic resilience indices based on (i) bounds on control variables, (ii) presence of right half plane transmission zeroes, (iii) presence of time delays, and (iv) plant-model mismatch. The proposed indices were utilized to improve the operability of numerous process, including heat integrated reactor networks [49, 50, 51], separation systems [52], heat exchanger networks [53].

Among the four aforementioned resilience indices, Perkins and Wong (1985) [54] studied the last two by adapting the "functional controllability" theorem proposed by Rosenbrock (1970) [55]. The authors further define a system to be functionally controllable if there exists a manipulated action $u(t)$ that can generate any process output $y(t)$ at any time t. Psarris and Floudas studied the dynamic resilience and functional controllability of multiple input multiple output (MIMO) closed loop systems with time delays [56, 57, 58], and transmission zeroes [57, 58]. Barton et al. (1991) [59] investigated the open loop process indicators, namely minimum singular value and right half plane zeros, to assess the interactions between different design configurations and their operability with the best possible control configurations.

In the context of simultaneously assessing the process controllability in process design, one of the first significant contributions is the "back-off approach" introduced by Narraway et al. (1991) [60]. Narraway and Perkins (1994) [61] used this approach to systematically assess the trade-offs between all possible controlled and manipulated variable pairs in a mixed integer formulation. Bahri et al. (1995) [62] employed the back-off approach to handle process uncertainties in an optimal control problem. The proposed approach is applicable to design linear and mildly nonlinear processes, and relies on three key steps, namely (i) perform a steady state nonlinear process optimization, (ii) linearize the process at the optimum point, and (iii) "back-off" from the optimal solution by some distance to ensure the feasibility of the operation under some structured disturbance profile. The proposed approach was shown to be effective effective to select between alternative flowsheets as well as alternative control structures.

With the burgeoning interest in exploring the simultaneous design and control problem, the In-

ternational Federation of Automatic Control (IFAC) organized the first workshop on "Interactions between Process Design and Process Control" in the Centre for Process Systems Engineering at Imperial College London in 1992. The workshop laid the groundwork for a plethora of approches with a wide range of diversity. Walsh and Perkins (1992) [63] implemented a PI loop in the flexibility analysis, where the input-output loop is selected by an exhaustive screening procedure. Luyben and Floudas (1992) [64] formulated a multiobjective MINLP problem to simultaneously consider the disturbance rejection capacity of the control loop through disturbance condition number and relative gain array to evaluate the interactions between the inputs and outputs of a MIMO system, while designing the process. Shah et al. (1992) [65] used the State-Task Network (STN) representation [66] to simultaneously consider the scheduling and design problems in a batch plant. Thomaidis and Pistikopoulos (1992) [67] introduced a framework to consider the design problem simultaneously with (i) the process flexibility through stochastic flexibility index, (ii) the effect of equipment failures to the overall performance by combined flexibility-reliability index, and (iii) the impact of equipment availability by combined flexibility-reliability index. These aforementioned novel approaches were shown to be promising concepts and techniques to address multiple facets of operational decisions simultaneously with the process design problem. As a result, succeeding studies after this workshop expanded these techniques and branched out to explore further opportunities.

Integrating PI controllers in the design optimization problem was one of the prominent outcomes of the workshop and became the most attractive option for the following research. The literature on PI controllers was already abundant and well-established by the time. Moreover, the explicit form of the controller structure made the integration relatively easy and intuitive, which significantly accelerated the research in closed loop design optimization. Walsh and Perkins (1994) [68] presented an integrated PI control scheme and process design for waste water neutralization. Although the proposed approach was effective for the SISO process, it was reported that it entails further challenges for more complex processes. One major drawback of PI control is its inability to tackle MIMO systems without any advanced modifications in the feedback loop structure. Nar-

raway and Perkins (1993, 1994) [69, 61] developed an MILP based formulation to systematically evaluate the economical performance of every input-output pair combination. Luyben and Floudas (1994a, 1994b) [70, 71] adapted a similar approach in a multiobjective framework to determine the best performing input-output pair based on the controllability indices introduced by them, earlier (1992) [64]. The proposed framework was showcased on the design of a heat integrated distillation system [70] and a reactor – separator – recycle system [71]. Mohideen et al. (1996) [35] formulated a multiperiod design and control problem to account for the dynamic variations in the operation, while including the input-output pairing superstructure in the problem. Moreover, the authors utilized the flexibility index to account for the uncertain parameters in the model and presented a decomposition algorithm for the resulting complex problem. Bansal et al. (2000) [72] constructed a similar formulation as a mixed-integer dynamic optimization (MIDO) problem, which was solved by a Generalized Benders Decomposition (GBD) based algorithm. The MIDO formulation was presented as follows.

$$\begin{aligned} \min_{u,des} \quad & \sum_{i \in NS} w_i C\big(\dot{x}^i(t), x^i(t), u^i(t), des^i\big) \\ s.t. \quad & \dot{x}^i(t) = h_d\big(x^i(t), u^i(t), des^i, \theta^i, t\big), x(t) = x_0 \\ & y^i(t) = h_a\big(x^i(t), u^i(t), des^i, \theta^i, t\big) \\ & g\big(\dot{x}^i(t), x^i(t), y^i(t), u^i(t), des^i, \theta^i, t\big) \leq 0 \end{aligned} \tag{1.6}$$

where w_i is the discrete probability of a scenario i and NS is the discretized set of scenarios. The discretization of uncertainty in the process was first proposed by Grossmann and Sargent (1978) [12].

Although the aforementioned PI based design and control frameworks are applicable on nonlinear processes, the range of operability is usually limited due to the mismatch between the nonlinear process model and the linearized control model. Ricardez Sandoval et al. (2008, 2009) [73, 74] used robust control tools and the back-off approach to integrate PI control and ensure its stability while solving the design optimization problem. The proposed approach was also tested against the

Tennessee Eastman process [75]. The back-off approach was later generalized for control structure selection in nonlinear processes by Kookos and Perkins (2016) [76]. Ricardez-Sandoval & co-workers have extensively studied back-off approach for simultaneous process design and control under uncertainty [77, 78, 79].

One main limitation of integrating PI control in the design optimization in a dynamic formulation is the increasing problem size and complexity. Kookos and Perkins (2001) [80] developed an algorithm for the integrated PI control and design optimization problem, where the size of the search space was reduced systematically in each successive iteration. Malcolm et al. (2007) [81] proposed an "embedded control optimization" procedure, where the authors introduced a two-stage decomposition scheme that approximates the complete integrated problem. The proposed approach reduced the problem size and complexity, and was showcased on larger scale problems including a reactor – separator system [82].

Apart from the inability to naturally handle MIMO systems, PI controllers do not explicitly account for any process constraints stemming from operational, environmental, and safety limitations. Model predictive control (MPC) overcomes these shortcomings by postulating a constrained dynamic optimization problem subject to an explicit model of the process [83]. One of the first remarkable efforts to integrate an MPC scheme in a nonlinear design problem was published by Brengel and Seider (1992) [84]. Here, the authors postulate a bi-level optimization problem, where the *leader* has an economic objective, while the *follower* is the MPC formulation, as presented by Eq. 1.7.

$$
\begin{aligned}
\min_{des} \quad & C_{des}\big(des\big) + \kappa C_{\kappa}\big(x(t), y(t), u(t), des, \theta(t)\big) \\
s.t. \quad & f_{des}\big(des, \theta(t)\big) = 0 \\
& g_{des}\big(des, \theta(t)\big) \leq 0 \\
& \min_{u(t)} \quad C_u\big(x(t), y(t), u(t), des, \theta(t)\big) \\
& s.t. \quad \dot{x} = f_u\big(x(t), y(t), u(t), des, \theta(t)\big) \\
& \qquad g_u\big(x(t), y(t), u(t), des, \theta(t)\big) = 0 \\
& \qquad h_u\big(x(t), y(t), u(t), des, \theta(t)\big) \leq 0
\end{aligned}
\tag{1.7}
$$

where κ is the design and control integration parameter that scales the trade-off between the controllability of the system and the investment cost. The bi-level problem presented in Eq. 1.7 is challenging to solve without appealing to simplifications. Therefore, the authors proposed replacing the follower problem by complementary slackness equations. However, the solution strategy was still intractable for more complex systems due to the numerical calculation of the second derivatives [84]. As a consequence, integration of the MPC scheme in the design optimization had been rather limited in the literature for almost a decade, until the invention of multiparametric MPC (mpMPC/explicit MPC).

Bemporad et al. (2002) [85] proposed formulating the MPC problem as an explicit function of the initial conditions of the system. This novel strategy allowed for deriving piecewise affine explicit control laws by treating the initial conditions as parameters. The proposed approach formulated the explicit MPC problem as presented by Eq. 1.8.

$$
\begin{aligned}
u_t(\theta) = \quad & \arg\min_{u_t} \|x_N\|_P^2 + \sum_{t=1}^{N-1} \|x_t\|_Q^2 + \sum_{t=1}^{N-1} \|y_t - y_t^{sp}\|_{QR}^2 + \sum_{t=0}^{M-1} \|u_t - u_t^{sp}\|_R^2 + \sum_{t=0}^{M-1} \|\Delta u_t\|_{R1}^2 \\
s.t. \quad & x_{t+1} = Ax_t + Bu_t + Cd_t, \quad y_t = Dx_t + Eu_t + Fd_t \\
& \underline{x}_t \leq x_t \leq \overline{x}_t, \quad \underline{y}_t \leq y_t \leq \overline{y}_t, \quad \underline{u}_t \leq u_t \leq \overline{u}_t, \quad \underline{\Delta u}_t \leq \Delta u_t \leq \overline{\Delta u}_t, \quad \underline{d}_t \leq d_t \leq \overline{d}_t \\
& \theta = [x_{t=0}, u_{t=-1}, d_t, y_t^{sp}, u_t^{sp}]^T
\end{aligned}
\tag{1.8}
$$

where N is the prediction horizon, M is the output horizon, superscript sp denotes set point, Q, QR, R, and $R1$ are the corresponding weight matrices determined by tuning, P is calculated by discrete algebraic Riccati equation, and $\|\cdot\|_\psi$ denotes weighted vector norm with a weight matrix ψ. Different than conventional MPC, Eq. 1.8 formulates the optimal control problem exactly and completely offline as a function of the set of parameters θ. The solution of this problem can be determined by multiparametric programming techniques, which express the solution space as a piecewise affine function, as presented by Eq. 1.9.

$$
\begin{aligned}
& u_t(\theta) = K_n\theta + r_n, \quad \forall\theta \in CR_n \\
& CR_n := \{\theta \in \Theta \mid CR^A\theta \leq CR^b\}, \quad \forall n \in \{1, 2, ..., NC\}
\end{aligned}
\tag{1.9}
$$

where CR_n is refered as a critical region and it is the active polyhedral partition of the feasible parameter space, Θ is a closed and bounded set, and NC is the number of critical regions.

The control law given by Eq. 1.9 reduces the complexity of solving an online optimization problem to a simple look-up table algorithm (also known as point location problem) and function evaluation, all of which are affine operations. Hence, the complexity of implementing an MPC scheme is similar to that of a PI controller.

Sakizlis et al. (2003) [86] exploited the explicit nature of the mpMPC solution in the context of design and control integration. The authors formulated a bi-level mixed integer dynamic optimization problem similar to Eq. 1.7, where the leader accounted for the investment and operating costs in the objective function subject to the dynamic high fidelity model, and the follower MPC

problem was substituted by the affine control law Eq. 1.9. The proposed formulation offered an elegant and systematic methodology to reduce the complexity of the bi-level Eq. 1.7 into a single level dynamic optimization problem. However, the solution strategy still required repetitive linearizations and solving a multiparametric programming problem at every iteration, which can be restrictive for large scale complex problems. Diangelakis et al. (2017) [87] alleviated that limitation by deriving a "design dependent offline controller", which allowed for solving a single MIDO problem after integrating the control law in the high fidelity model. Eliminating the linearization step and formulating a single synergistic design and control problem also improved the economical performance of the resulting process compared to the approach proposed by Sakizlis et al. (2003) [86]. The proposed formulation was also showcased on a tank, a continuous stirred tank reactor, and a residential scale combined heat and power unit. The cost effectiveness of the MPC integrated optimal design was also reported to be superior than PI integrated approaches in the literature. Diangelakis and Pistikopoulos (2017) [88] reported that the mpMPC integrated optimal combined heat and power unit operated more fuel efficient in closed loop than PI integrated design. Similarly, Sanchez-Sanchez and Ricardez-Sandoval (2013) [89] showcased a system of CSTRs, where the MPC integrated framework reduced both the operating and the investment costs compared to the PI control integrated approach.

One common aspect of the studies on simultaneous design and control optimization is the assumption that the process will be operated around the same steady-state point throughout the entire life cycle of the plant. However, the external plant conditions, such as market conditions, may dictate a considerably wider operating region with multiple steady state points [3]. The increasing competition among the businesses impacts the volatility of the market, which creates rapid fluctuations in the energy and raw material prices as well as their availability. Moreover, the demand rate on the product is also subject to considerable variations during the plant operation. Therefore, it is clear that there exists a "best" operating strategy under the knowledge available to the operator, which necessitates the operability of the plant across a wider range. For example, high production rates may be less profitable during the night time because of increased energy prices

and hence, operating the energy intensive processes during the day time may reduce the operating costs. This indicates that the operating level of a processing unit might vary drastically by the choice of the operator. However, the integrated design and control frameworks discussed in this section usually assume a single operating point around which a controllability and flexibility analysis is conducted. Consequently, these frameworks do not attempt to provide any means of guaranteeing the operability of the process at different regions. In the next section, we will discuss several approaches that account for multiple operating regions in a plant, and their scheduling during the operational optimization.

1.3 Towards the grand unification of process design, scheduling, and control

Process design, scheduling, and control problems are traditionally constructed to address different objectives and they span widely different time scales. In a nutshell, the plant design problem dictates the capacity of processing and it usually comprises the most uncertainty due to its years long life-cycle. The scheduling problem addresses the allocation of the resources and time, as well as the operating level of processing units and their maintenance based on some economical criteria over days/months long horizons. Lastly, the control problem maintains the performance of the plant, while satisfying any physical limitations such as the environmental and safety constraints. The discrepancy in the objectives and time scales creates a challenging problem to systematically evaluate and determine the optimal trade-off between different decision makers.

Process scheduling is more critical in batch operations than continuous operations, as the former are inherently dynamically operated. Accordingly, the initial efforts focused primarily on the batch processes for the integration of the operational optimization and design problems. Birewar and Grossmann (1989) [90] formulated NLP models to incorporate the scheduling decisions in the batch sizing and timing problem in a multiproduct plant for unlimited intermediate storage and zero wait policies. Shah et al. (1992) [65] tackled a similar problem by using the STN representation. White et al. (1996) [91] investigated the switchability of continuous processes between different operating points through formulating an optimal control problem that accounts for the terminal criteria and path constraints within a range of design parameters. Bhatia and Biegler (1996, 1997)

[92, 93] formulated a dynamic optimization problem, where an economic objective function was subject to a dynamic high fidelity model of the process described by differential algebraic system of equations. The authors proposed a solution strategy based on discretizing the process model by orthogonal collocation over finite elements, followed by solving the resulting NLP by using a standard solver. The proposed modeling and solution strategy was shown to be promising to satisfy the path constraints, which is a crucial benefit for dynamic systems. Terrazas-Moreno et al. (2008) [4] extended this integration approach to account for the binary decisions in the scheduling problem by formulating a MIDO. Similar to Bhatia and Biegler (1996, 1997) [92, 93], the authors first discretized the problem by orthogonal collocation, followed by solving the resulting MINLP.

The early studies that explore the interactions between the scheduling and process control decisions have a significant role in shaping today's approaches for the integrated design optimization problem. In their excellent review article, Baldea and Harjunkoski (2014) [94] classified these attempts to integrate the scheduling and control decisions as (i) "top-down approaches", where the process dynamics and control elements are incorporated in a scheduling skeleton, and (ii) "bottom-up approaches", where the process economics are implemented in the plant-wide control decisions.

In terms of characterizing the transitions between different products in a single operating unit, Mahadevan et al. (2002) [95] introduced a unique "top-down" perspective on the operational optimization problem, revealing that a simultaneous approach on the scheduling and control problem can identify and eliminate the fundamental limiting behavior during the transitions, as showcased on a polymer grade transition process. However, the presented approach requires case specific heuristic decisions to select the "best" fitting scheduling and control configuration and hence, it is not suitable for different applications in the general sense. Chatzidoukas et al. (2003) [96] studied a similar polymerization reactor, and formulated a MIDO problem to determine the time optimal transition between different polymer grades and best performing control structure simultaneously. Flores-Tlacuahuac and Grossmann (2006) [97] introduced a monolithic approach on a multiproduct cyclic CSTR, where the profit was maximized by manipulating the production sequence, transition times, production rates, length of processing times, and amounts manufactured

of each product. Different from the earlier studies [95, 96], the authors focused on the manipulated actions rather than the optimal control configuration. They formulated a MIDO problem, which was solved by discretization of the differential algebraic equations by orthogonal collocation on finite elements followed by solving the resulting MINLP. The presented approach has been extensively studied in the following years to broaden its scope and effectiveness. Terrazas-Moreno et al. (2007) [98] applied this approach on two industrial polymerization reactors. Terrazas-Moreno et al. (2008) [4] formulated a design optimization problem accounting for the scheduling and open loop control trajectories using this approach. Flores-Tlacuahuac and Grossmann (2010, 2011) extended the formulation to partial differential equation systems, and showcased on tubular reactors with single [99] and multiple production lines [100].

This monolithic approach usually generates open loop control trajectories, i.e. no feedback loop is assumed to develop the input and output profiles. However, the processing units are subject to internal process disturbances, and the mismatch between the process and the model leads to deviations in the targeted operations. Zhuge and Ierapetritou (2012) [101] implemented the monolithic approach in closed loop, where the authors initiate a readjustment procedure to solve the integrated problem online if the states deviate from their reference trajectories. This approach does not completely resolve issue of handling the process disturbances or the process/model mismatch, however it was shown to mitigate these concerns to a great extent. Gutiérrez-Limón et al. (2014) [102] also implemented a similar closed loop strategy with a nonlinear model predictive control scheme, while extending the scope of the problem statement to account for an extended horizon production policy. However, both approaches require solving a complex and large scale MINLP problem at the time steps of the controller, which makes it unsuitable for the processes with fast dynamics.

Low-order representation of fast process dynamics in the scheduling problem has been an effective approach to reduce the computational burden of solving complex optimization problems. Du et al. (2015) [103] proposed a time scale-bridging model that describes the closed-loop input–output behaviour of a process in the scheduling formulation, postulated as a MIDO problem.

The low-order representation also maintains the stability of the process in the existence of process/model mismatch and handles disturbances. Baldea et al. (2015) [104] extended this approach to MPC governed systems.

Burnak et al. (2018) [105] also addressed the online computational burden of "top-down" approaches by developing a multiparametric programming based approach, where the authors explicitly mapped (i) the closed loop dynamic process behavior in a "control-aware" scheduling problem, and (ii) the continuous and binary scheduling level decisions such as the operating level and operational mode of the system in a "schedule-aware" MPC scheme (iii) to yield the optimal operational decisions. The offline nature of the integrated scheduling and control scheme allows for determining the feasible operating space prior to actualizing the operation. Furthermore, reducing the problem complexity from solving online optimization problems to a simple look-up table and affine function evaluation, the framework is well-suited for fast process dynamics. Charitopoulos et al. (2019) [106] employed a similar multiparametric programming approach to include the planning decisions in their framework.

In the "bottom-up" approaches, on the other hand, incorporating the economical objectives in the plant control structures has been perceived as the key for seamless integration of scheduling and control. For this purpose, MPC formulations provide the flexibility to account for a spectrum of objectives in the control level due to their optimization based structures. Loeblein and Perkins (1999) [107] presented an economical analysis of unconstrained MPC scheme operating under constrained systems. The authors determined the most cost effective model predictive regulatory control structure by utilizing the back-off approach to satisfy the constraints. Zanin et al. (2002) [108] addressed the discrepancy between the real-time optimization (RTO) and control layers by incorporating the economic optimization problem in the controller and feeding the same piece of information in both layers. The proposed formulation diminishes the discrepancy between the decision layers to yield more economical operations, but the resulting control scheme does not guarantee the stability of the process for the entirety of operations. Rawlings and Amrit (2009) [109] developed asymptotic stability criteria by formulating the so-called "economic MPC" (or

EMPC), where the objective function of the MPC is designed to minimize the operational costs instead of maintaining the steady state of the process. This approach aims to replace the conventional two layer structure with RTO and dynamic regulatory control by a single control layer, where the economic optimization and process regulation are conducted simultaneously. Amrit et al. (2011) [110] further extended the stability criteria by (i) imposing a region constraint on the terminal state instead of a point constraint, and (ii) adding a penalty on the terminal state to the regulator cost.

Similar to the monolithic "top-down" scheduling and control approach, EMPC has been shown to be too complex to be solved in the control time steps. This limitation has led the researchers to develop decomposition algorithms for faster computational times. Würth et al. (2011) [111] proposed a decomposition framework for the single layer dynamic RTO formulation, where the slow trends and process uncertainty is handled in the upper layer, while the lower layer accounts for the fast disturbances actiong on the process. Ellis and Christofides (2014) [112] focused on selecting a suitable input configuration for such two-layered dynamic RTO structures such that the asymptotic stability is guaranteed. Jamaludin and Swartz (2017) [113] and Li and Swartz (2019) [114] employed a convex MPC problem in the lower level regulatory control, which enabled its exact substitution with KKT optimality conditions. Simkoff and Baldea (2019) [115] used the same substitution strategy on a production scheduling problem.

Design optimization accounting for the scheduling and control decisions with closed loop implementation is relatively recent in the literature. Patil et al. (2015) [5] modeled the product transitions in design optimization, while maintaining the stability of the closed loop system governed by a PI control scheme. The authors formulated an MINLP similar to Eq. 1.6 with the contribution of the criterion, $eig(A_i^z(x_{lin})) < 0$, which enforces the stability of the linearized states for all products i in a multiproduct unit under all critical scenarios z. Due to the linearization of the controllers around the operating point, this approach requires repetitive identification of the states at every optimization iteration. Koller and Ricardez-Sandoval (2017) [6] improved this approach by applying orthogonal collocation on finite elements on the integrated problem, and Koller et al. (2018) [7] employed the back-off method to satisfy the constraints under uncertainty by using Monte Carlo

sampling techniques to determine the back-off terms.

Recently, Burnak et al. (2019) [8] introduced a multiparametric programming based theory and framework for the integration of process design, scheduling, and control, which will be detailed in the remainder of this thesis. To summarize, we derive offline design dependent control and scheduling schemes that can be incorporated in a MIDO formulation in a multi-level fashion, as presented by Eq. 1.10.

$$
\begin{aligned}
\min_{u,s,des} \quad & \int_0^{\tau} C(x(t), y(t), u(t), s(t), des, d(t))dt \\
s.t. \quad & \dot{x}(t) = f(x(t), y(t), u(t), s(t), des, d(t), t) \\
& \underline{y} \leq y(t) = g(x(t), y(t), u(t), s(t), des, d(t), t) \leq \overline{y} \\
& \underline{x} \leq x(t) \leq \overline{x}, \quad \underline{des} \leq des \leq \overline{des}, \quad \underline{d} \leq d(t) \leq \overline{d} \\
& s_t(\theta_s) = \arg\min_{s} \quad \sum_{t_s \in N_s} C_s(x_{t_s}, y_{t_s}, s_{t_s}, des, d_{t_s}) \\
& \qquad s.t. \quad \underline{x}_{t_s} \leq x_{t_s+1} = A_{t_s} x_{t_s} + B_{t_s} s_{t_s} + C_{t_s} d_{t_s} \leq \overline{x}_{t_s} \\
& \qquad\quad \underline{y}_{t_s} \leq y_{t_s} = D_{t_s} x_{t_s} + E_{t_s} s_{t_s} + F_{t_s} d_{t_s} \leq \overline{y}_{t_s} \\
& \qquad\quad \underline{s}_{t_s} \leq s_{t_s} \leq \overline{s}_{t_s}, \quad \underline{d}_{t_s} \leq d_{t_s} \leq \overline{d}_{t_s} \\
& \qquad\quad \underline{\theta}_s \leq \theta_s = [x_{t_s=0}^T, y_{t_s=0}^T, d_{t_s}, des]^T \leq \overline{\theta}_s \\
& u_t(\theta_c) = \arg\min_{c} \quad \sum_{t_c \in N_c} C_c(x_{t_c}, y_{t_c}, u_{t_c}, des, d_{t_c}) \\
& \qquad s.t. \quad \underline{x}_{t_c} \leq x_{t_c+1} = A_{t_c} x_{t_c} + B_{t_c} u_{t_c} + C_{t_c} d_{t_c} \leq \overline{x}_{t_c} \\
& \qquad\quad \underline{y}_{t_c} \leq y_{t_c} = D_{t_c} x_{t_c} + E_{t_c} u_{t_c} + F_{t_c} d_{t_c} \leq \overline{y}_{t_c} \\
& \qquad\quad \underline{u}_{t_c} \leq u_{t_c} \leq \overline{u}_{t_c}, \quad \underline{d}_{t_c} \leq d_{t_c} \leq \overline{d}_{t_c} \\
& \qquad\quad \underline{\theta}_c \leq \theta_c = [x_{t_c=0}^T, y_{t_c=0}^T, d_{t_c}, des]^T \leq \overline{\theta}_c
\end{aligned}
\tag{1.10}
$$

where s and u denote the scheduling and control decisions, respectively. Note that the proposed formulation postulates explicit expressions for the scheduling and control strategies as functions of a set of parameters, θ, which includes the design of the process. The design dependence of the

operational strategies allows for their direct integration in the MIDO formulation. The postulated formulation has two main benefits, (i) due to the explicit form of the follower problems, the multi-level MIDO problem is reduced to a single level, and (ii) only the design variables are left as the degrees of freedom of the problem, since the remaining are determined as a function of the design.

1.4 Current challenges and future directions

The PSE community has achieved unequivocally remarkable progress in realizing and advancing the set goals of Professor Sargent on systematic design optimization in five decades. Today, using design optimization tools to at least some extent has long become the standard practice in many industries. Commercial modeling and simulation software tools such as gPROMS[2] and Aspen Plus Dynamics[3] have been featuring robust and efficient solvers for dynamic optimization problems for a few years. Despite these milestones in PSE, we still have to make significant assumptions and simplifications regarding the operational decisions in the process design phase, even though the impact of their interdependence on process economics and operability has been articulated in numerous studies. Hence, the academia still needs to mature the theoretical foundations and the applicability of unified design optimization approaches before it gains wide industrial recognition. Here, we discuss some of the bottlenecks and potential directions to improve the state-of-the-art for industrial practice.

1.4.1 The need for an industrial benchmark problem

As we have presented in this paper, there is a plethora of proposed modeling techniques and solution approaches for the next generation unified design optimization problems. Therefore, it is clear that we need a generally accepted benchmark problem, preferably in industrial scale, to validate the effectiveness of proposed methodologies. The PSE community has benefited greatly from such standardized problems, such as the famous Tennessee Eastman Process detailed by Downs and Vogel (1993) [116] for process control studies. We believe that a well-defined problem will clarify the objectives in unified design optimization and accelerate the research towards industrial

[2]https://www.psenterprise.com/products/gproms
[3]https://www.aspentech.com/en/products/pages/aspen-plus-dynamics

expectations. The problem should describe at least the following.

i. *A high fidelity model that describes the dynamics of the process*. The model should feature appropriate design variables to exhibit the dynamic consequences of scaling up/down the process. Furthermore, considering the reduction in capital investment that the multipurpose and multiproduct operating units provide, the process should comprise such units to examine the scheduling/design and scheduling/control trade-offs. Recent research that consider process design, scheduling, and closed loop control problems simultaneously [8, 5, 7] have studied only a single processing unit, which reflects a limited fraction of the overall benefit that the grand unification can provide.

ii. *Cost relations for investment, utility, and raw materials*. A functional form of the investment cost with respect to the capacity of the process is required to have standardized comparable results. Also, utility costs and raw materials may vary significantly, which inevitably impacts the optimal scheduling decisions. For instance, grid electricity costs are known to exhibit considerable differences during the day and night times. Thus, operational loads in energy intensive processes may fluctuate heavily. The impact of such changes in operating levels on design and control decisions were discussed in Section 1.2.

iii. *Product demand and availability of the utility, raw materials, and operating units over a time horizon*. Production allocation and timing is a key aspect of scheduling problem, which are heavily dictated by the product demand and availability of resources. However, it is not a trivial practice to estimate the future of these quantities. Therefore, probability distributions of these components will be beneficial to determine their expected values, while being able to take into account their worst case scenarios.

1.4.2 Robust advanced control and scheduling strategies

Incorporation of advanced control schemes seamlessly in the design optimization problem requires the controller to capture the dynamics of the process for the entire range of design variables.

Burnak et al. (2019) [8] attempted to approximately model the design configuration as a right hand uncertainty in the constraint set, validated by closed loop simulations and closed loop MIDO problems. However, the design variables impose uncertainty in the left hand side of the constraints, as well as the nonlinear and bilinear terms in the objective function. Therefore, robust control strategies need to be developed for accurate predictions of future states in the control level prior to the realization of the design, and to guarantee the stability of the closed loop operations in simultaneous approaches.

Analogously, scheduling schemes should be robustified in the design optimization to minimize the rescheduling due to unexpected disruptive events, such as unit failure, drastic changes in product demand rate and raw material availability. Excluding these events in the scheduling scheme may result in steep changes in the target operation, and thus unattainable set points for the controller.

1.4.3 Considering flowsheet optimization, process intensification, and modular design opportunities

Optimization based plant design techniques have been used and developed for more than four decades [117, 118]. These techniques postulate "superstructures" that systematically simulate and compare every combination of flowsheet possibilities to determine the optimal process. More recently, superstructures have been formulated at the phenomena level to capture the fundamental relations between the mass and energy, which in turn yields intensified processes [119, 120, 121, 122, 123, 124, 125]. Such intensified processes are expected to deliver significantly increased operational efficiency and decreased unit volumes, making them very attractive options both in academia and industry [126]. This rapidly growing interest in intensified processes is one of the most pronounced directions that the PSE community has been taking. Therefore, studying these intensified processes in the context of unified design optimization will attract a wider audience from the industry. Clearly, modeling the spatial (synthesis/intensification) and temporal (scheduling/control) decisions simultaneously in a single problem formulation will capture even more synergistic interactions, which will increase the process profitability.

Furthermore, the researchers studying process intensification can benefit from the tools and methodologies on unification of design, scheduling, and control. Baldea (2015) [127] reported a theoretical justification for the loss of control degrees of freedom due to process intensification, which poses a significant limitation on intensification activities. Tian and Pistikopoulos (2019) [128] and Dias and Ierapetritou (2019) [129] discuss the limitations on the operability of such intensified systems and potential directions to overcome these limitations in their excellent review papers. The researchers on process intensification technologies can adopt the techniques, ranging from steady state and dynamic flexibility to integration of scheduling and control decisions, in order to address the operability issues.

1.4.4 Theoretical and algorithmic developments in MIDO

The most limiting bottleneck of the simultaneous approaches is the size of the integrated MIDO problems. The time component of the problem significantly increases the computational complexity, yielding infinitely many NP-hard problems to acquire an optimal solution profile. However, tailored algorithms can be developed by utilizing the special structure of such integrated problems. For instance, the open loop design optimization problem is relatively simpler than the integrated MIDO, and constitutes a lower bound on the optimal solution of the overall problem. Such properties can be exploited in decomposing the MIDO into subproblems to significantly reduce the search space for faster algorithms.

1.4.5 Software development

Despite the theoretical and practical advances in the unified design problem among the academia, there is no commercially available platform or a software prototype. Such a tool will make the integrated approaches more accessible to the process designers in industry who are not necessarily experts on process control and scheduling, and it will attract more researchers from different disciplines and backgrounds. Pistikopoulos et al. (2015) [130] introduced the PARametric Optimization & Control (PAROC) framework to design explicit controllers based on high fidelity models, which can be a viable option to address the grand unification challenge [8, 105, 131, 2, 88]. However, it

is clear that more progress is needed to engage a wider audience.

1.5 Scopse of this study

In this thesis, we consider a generic process where the interactions between the design (long term decision), schedule (middle term decision), and control (short term decision) problems are sufficiently significant to impact the feasibility and the optimality of each individual decision. Therefore, we define the following problem statement that encapsulates all three decisions simultaneously.

(i) *Given*: A high fidelity model based on first principles or data-driven modeling techniques that accurately captures the dynamics of the system, any physical limitations of the system due to process safety considerations or product specifications, unit costs for design, raw material, energy, and inventory, revenue for unit product, and an accurate demand forecast.

(ii) *Determine*: Production sequence throughout an operating horizon, closed loop control strategy that delivers the product specifications, set points for the operation tailored for the dynamics of the closed loop strategy, size of the processing equipment that ensures operability of the process.

(iii) *Objective*: Minimize the operating and capital costs.

Note that the objective of the problem can be replaced by the minimization of the energy utilization, CO_2 emissions, processing time, or a combination of these tasks based on the application without changing the framework. In this work, we showcase the minimization of costs as it is the most frequently used objective in process operations.

The generalized mathematical formulation of the defined problem statement is given by Eq. 1.11 in the form of a mixed-integer dynamic optimization problem [8].

$$
\begin{aligned}
\min_{u,s,des} \quad & J = \int_0^{\tau} P(x,y,u,s,des,d)dt \\
s.t. \quad & \dot{x} = f(x,u,s,des,d), \quad x(0) = x_0 \\
& \underline{y} \leq y = g(x,u,s,des,d) \leq \overline{y} \\
& \underline{u} \leq u = h(x,u,s,des,d) \leq \overline{u} \\
& \underline{s} \leq s = m(x,u,s,des,d) \leq \overline{s} \\
& \underline{x} \leq x \leq \overline{x}, \quad \underline{des} \leq des \leq \overline{des}, \quad \underline{d} \leq d \leq \overline{d}
\end{aligned}
\tag{1.11}
$$

where x are the states of the system, y are the system outputs, u are the control actions, s are the scheduling decisions, des are the design variables, and d are the measured disturbances, P is the cost function accounting for the operating and capital costs, f and g are differential and algebraic relations, h and m are the implicit relations that describe the operational decisions, and lower and upper bars are the bounds on the variables. We also differentiate the disturbances at the control level, $d^c \subseteq d$, such as the variations in the feed conditions, and the disturbances at the scheduling level, $d^s \subseteq d$, such as the fluctuating market prices and demand rates. Note that discrete design and scheduling decisions such as the number of trays in a distillation column, the product to be manufactured at a particular time instance, and resource allocation render Problem 1.11 a mixed-integer optimization problem.

The problem statement defined in Eq. 1.11 suggests that the high fidelity model given by f and g, the cost function P, the bounds on the variables are known, and a realistic demand scenario is available. The goal is to minimize the objective P over a time horizon τ by manipulating the degrees of freedom of the system available in the long term (des), middle term (s), and short term (c).

1.6 Research objectives and thesis outline

The general formulation presented by Eq. 1.11 is typically a large scale, non-convex dynamic optimization problem with integer variables, and is challenging to solve even for small scale problems comprising a single processing unit. Therefore, the objectives of this work to advance the

current knowledge on integration of multi-scale decisions are listed as follows.

Objective I Develop a process agnostic and synergistic framework to integrate the process design, scheduling, and control problems based on a single high-fidelity model. This goal requires a tractable form of the integrated process design, scheduling, and control problem, i.e. Eq. 1.11. Furthermore, addressing the operability of the process is another challenge, since the "optimal" profile determined by solving Eq. 1.11 without the consideration of the scheduling and control decisions may not be a viable path during the online implementation. Here, multiparametric programming is vital with its ability to map the optimal receding horizon policies explicitly, which can be embedded exactly in the generalized MIDO formulation.

Objective II Investigate the interactions between the process design and control problems under a single operating window. Based on their inherent relationship, (i) develop a single optimal control strategy that is applicable under a range of design configurations, and (ii) determine the optimal process design considering the closed-loop dynamics introduced by the controller.

Objective III Investigate the interactions between the scheduling and control problems for a given design configuration. Develop (i) a single optimal control strategy that is applicable under a range of operating regions, and (ii) a scheduling strategy that is aware of the closed-loop dynamics of the system.

Objective IV Develop an modeling technique and optimization approach to efficiently incorporate the operational strategies in an MIDO formulation. Such an approach is especially vital for the integration of decisions in batch processes, where the inherently dynamic systems and the high number of discrete decisions increase the problem complexity compared to the continuous processes.

Objective V Develop an algorithm to solve large scale multiparametric programming problems, which is an inevitable result of integrating the control and scheduling problems in a single formulation.

2. INTEGRATION OF PROCESS DESIGN AND CONTROL[1]

2.1 Introduction

Traditionally, chemical plants have been designed with sequential assessment of the process design based on the steady state economics, followed by a heuristically tuned control strategy to govern the real-time operations around the designated steady-state. The independent characterization of these two aspects (i) hinders the feedback controllers to respond better to disturbances and process uncertainty, and (ii) results in overdesigned process equipments which directly increase the capital costs, and impact the operating costs [132].

Considering the strong interactions between the design and control problems, a plethora of approaches has been proposed in the past three decades. An indicative list of the contributions proposed on the literature is provided in Table 2.1 [8]. The earlier approaches primarily focus on developing *a priori* metrics for the reliability of the steady state design under process disturbances and uncertainties based on feasibility, flexibility, stability, and controllability of the system. The advances in computational power and mathematical programming techniques has shifted the recent studies in the direction of simultaneous consideration of the control strategy during the process design phase [88].

In this chapter, a process agnostic framework is presented to simultaneously address the process design and control problems via multiparametric programming. A design dependent control strategy is developed by using the PARametric Optimization and Control (PAROC) framework [130] to derive an offline map of control actions as an explicit function of the design variables. The offline map allows for direct implementation of the control strategy in the design optimization problem. The integrated problem corresponds to an MIDO, given by Eq. 1.11, where the scheduling decisions s are assumed to be constant.

[1]Portions of this chapter have been adapted from Diangelakis, N.A., Burnak, B., Katz, J., Pistikopoulos, E.N., Process Design and Control optimization: A simultaneous approach by multi-parametric programming, AIChE Journal 2017, 63 (11), pp. 4827-4846 and Burnak, B., Diangelakis, N.A., Katz, J., Pistikopoulos, E.N., Integrated process design, scheduling, and control using multiparametric programming, Computers & Chemical Engineering 2019, 125, pp. 164-184, *Special Issue*, with permission from Elsevier.

Table 2.1: Design and control in the literature: An indicative list (Reprinted with permission from [88]).

Author (year)	Contribution
Perkins & co-workers (1991) [60], Kravaris & co-workers (1993) [133, 134], Pistikopoulos & co-workers (1994, 1997) [135, 37], Floudas & co-workers (1994, 2000, 2001) [70, 136, 137], Romagnoli & co-workers (1997) [138], Skogestad & co-workers (2014) [139], Gani & co-workers (1995) [140], Francisco & co-workers (2014) [141]	Feasibility, flexibility, stability, controllibility, resilience metrics in steady-state design optimization with MIDO or MINLP
Pistikopoulos & co-workers (2000, 2002, 2003a, 2003b, 2004) [72, 142, 143, 144, 145, 86, 146], Linninger & co-workers (2007) [81], Swartz & co-workers (2014) [147], Ricardez-Sandoval (2012) [148]	Integrated MIDO formulation/ decomposition with PID control or (mp)MPC
Biegler & co-workers (2007, 2008) [149, 150], Seider & co-workers (1992) [84], Ricardez-Sandoval & co-workers (2008, 2016, 2017)[73, 77, 78], Pistikopoulos & co-workers (1996) [36], Perkins & co-workers (2002, 2016) [151, 76], Flores-Tlacuahuac & co-workers (2009) [152], Barton & co-workers (2011, 2015) [153, 154, 155], Linninger & co-workers (2006) [156]	Iterative MINLP formulation with stochastic back-off formulation for uncertainty
Francisco & co-workers (2014) [157], Ricardez-Sandoval & co-workers (2009, 2011) [132, 158], Gani & co-workers (2012) [159], Mitsos & co-workers (2014) [160]	Review articles on design and control integration

The remainder of this chapter is organized as follows. In Section 2.2, the PAROC framework, its utilization to derive design-dependent controllers, and the formulation of the integrated MIDO problem are discussed in detail. An example comprising a binary distillation column is presented to showcase the introduced framework in Section 2.3. Finally, the conclusions are summarized in Section 2.4.

2.2 Simultaneous design and control optimization via PAROC

The PAROC framework and software platform provides a comprehensive environment to design chemical processes, to build advanced controllers, and to perform parameter estimation based on high-fidelity models benefiting from the most recent advances in the filed of multiparametric prgramming [130]. The first step of the proposed methodology is to acquire a mathematical "high-fidelity" model to describe the system of interest with sufficient accuracy. These models are typically very large in size and complex in nature, rendering it difficult to directly apply an advanced control strategy. Therefore, the original mathematical model is approximated or reduced in size via the existing algorithms in the literature. Note that the design variables should be maintained in the lower dimensional model to lay the foundation of a design dependent controller. A model predictive control (MPC) scheme is constructed using the approximate model, and solved multiparametrically (mpMPC) to generate offline maps of optimal control actions as a function of the design variables. The developed design dependent explicit expressions for the optimal control actions are embedded exactly in the dynamic high-fidelity model, allowing for a dynamic optimization problem to be formulated to determine the optimal design configuration and the control strategy simultaneously. The steps of the framework are illustrated in Figure 2.1, and detailed as follows.

Step 1 – High-fidelity dynamic modeling: A rigorous mathematical model based on first principles, typically differential algebraic equations (DAE) or partial differential algebraic equations (PDAE), is used to simulate the dynamics of the system. In this work, we use the gPROMS® environment to construct the model, as described with a general representation in Eq. 2.1.

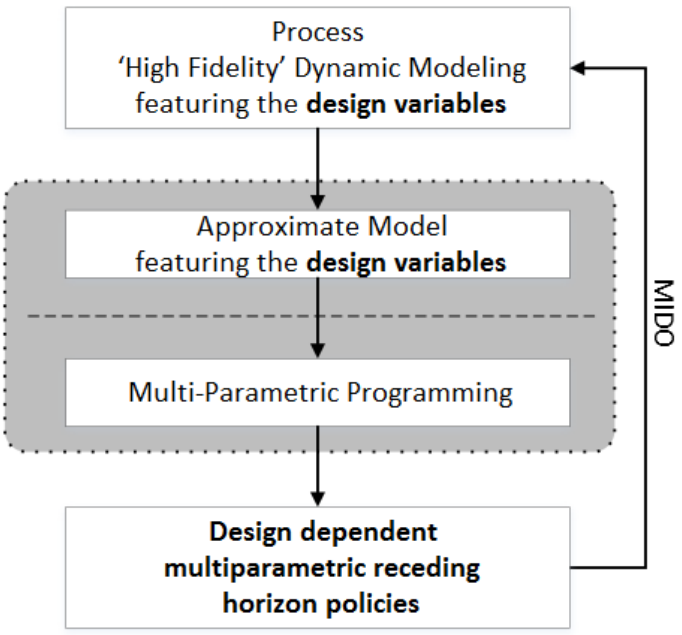


Figure 2.1: The PAROC framework approach for simultaneous design and control. Actions within the gray area happen once and offline (Reprinted with permission from [88]).

$$\begin{aligned} \dot{x}(t) &= f(x(t), u(t), d(t), des, t) \\ y(t) &= g(x(t), u(t), d(t), des, t) \end{aligned} \tag{2.1}$$

where x are the states, u are the manipulated variables, d are the external disturbances, des are the design variables, t is independent time variable, y are the system outputs, and f and g are generic functions.

Step 2 – Model approximation: The high-fidelity model designed in Step 1 usually comprises highly nonlinear and non-convex terms that renders the practice of advanced control algorithms quite challenging. Therefore, we appeal to system identification or model reduction techniques to approximate the model formulation with a discrete time affine state space representation. In this work, the model approximation is performed via the MATLAB® System Identification Toolbox™ yielding the state space representation in Eq. 2.2. Note that the system identification techniques may or may not preserve the physical meanings of the original states in Eq. 2.1.

$$
\begin{aligned}
x_{t_c+1} &= A\, x_{t_c} + B\, u_{t_c} + C\, [d_{t_c}^T, des_{t_c}^T]^T \\
\hat{y}_{t_c} &= D\, x_{t_c} + E\, u_{t_c} + F\, [d_{t_c}^T, des_{t_c}^T]^T
\end{aligned}
\tag{2.2}
$$

where A, B, C, D, E, and F are matrices with appropriate dimensions, $\hat{y}$ are the outputs predicted by the controller, and t_c is the discretization step of the state space model. Addressing the plant-model mismatch in the control problem will be addressed in the following step.

Note that the design variables are treated as a disturbance and thus, cannot be manipulated in the state space model. Inclusion of the design variables is vital to represent the range of dynamics of the system of interest in the control strategy.

Step 3 – Multiparametric model predictive control (mpMPC): The design of the feedback controllers is based on the procedure described by Bemporad et al., (2002) [85]. The generalized form of the mpMPC problem is described by Eq. 2.3.

$$
\begin{aligned}
u_{t_c}(\theta) = \arg\min_{u_{t_c}} \quad & \|x_{N_c}\|_P^2 + \sum_{t_c=1}^{N_c-1} \|x_{t_c}\|_Q^2 + \sum_{t_c=1}^{N_c-1} \|y_{t_c} - y_{t_c}^{SP}\|_{QR}^2 \\
& + \sum_{t_c=0}^{M_c-1} \|u_{t_c} - u_{t_c}^{SP}\|_R^2 + \sum_{t_c=0}^{M_c-1} \|\Delta u_{t_c}\|_{R1}^2 \\
s.t. \quad & x_{t_c+1} = Ax_{t_c} + Bu_{t_c} + C[d_{t_c}^T, des^T]^T \\
& \hat{y}_{t_c} = Dx_{t_c} + Eu_{t_c} + F[d_{t_c}^T, des^T]^T \\
& y_{t_c} = \hat{y}_{t_c} + e \\
& e = y_{t_c=0} - \hat{y}_{t_c=0} \\
& \underline{x} \le x_{t_c} \le \overline{x}, \quad \underline{y} \le y_{t_c} \le \overline{y} \\
& \underline{u} \le u_{t_c} \le \overline{u}, \quad \underline{\Delta u} \le \Delta u_{t_c} \le \overline{\Delta u} \\
& \theta = [x_{t_c=0}^T, u_{t_c=-1}^T, d_{t_c=0}^T, des^T]^T \\
& \forall t_c \in \{0, 1, ..., N_c - 1\}
\end{aligned}
\tag{2.3}
$$

where θ is the set of bounded parameters, N_c is the output horizon, M_c is the control horizon, $\|\cdot\|_\Psi$ denotes weighted vector norm with a weight matrix Ψ, SP denotes set point, P, Q, QR, R, $R1$

are the corresponding weight matrices. An error term e denotes the mismatch between the actual system output and the predicted output at initial time.

The resulting multiparametric programming problem is solved via the POP toolbox [161] in the MATLAB environment, yielding the explicit map of optimal control actions as a function of the design variables, as given by Eq. 2.4.

$$u_j(\theta) = K_n\theta + r_n, \forall\theta \in CR_n \tag{2.4a}$$

$$\theta := \left[x^T_{t_c=0}, u^T_{t_c=-1}, d^T_{t_c=0}, des^T, (y^{SP}_{t_c})^T, (u^{SP}_{t_c})^T, y^T_{t_c=0}\right]^T \tag{2.4b}$$

$$\forall j \in \{0, 1, ..., M_c\}, \forall t_c \in \{0, 1, ..., N_c\} \tag{2.4c}$$

where an arbitrary n^{th} critical region CR_n is defined as a closed polyhedron given by $CR_n := \{\theta \in \Theta \mid L_n\theta \leq b_n\}$. Equation 2.4 defines a piece-wise affine optimal control law over the parameter space comprising the states $x_{t_c=0}$, control action at the previous time step $u_{t_c=-1}$, measured external disturbance $d_{t_c=0}$, design variables des, output set points $y^{SP}_{t_c}$, input trajectories $u^{SP}_{t_c}$, and output measurements $y_{t_c=0}$.

Remark 1. *Inclusion of the design variables in the optimal control law allows for the use of a single control strategy across a range of design configurations.*

Step 4 – Closed loop validation: The derived control law, Eq. 2.4 is validated against the dynamic high fidelity model Eq. 2.1 within the bounds of design variables. Here, the control law is expected to deliver satisfactory performance by (i) achieving effective set point tracking, (ii) satisfying the path constraints, and (iii) maintaining stability. If the controller fails to deliver sufficient performance, either the weight matrices in Eq. 2.3 are tuned accordingly, or the model approximation step is revisited to develop a more accurate open loop low order model.

Step 5 – Dynamic optimization: The optimal control law, i.e. Eq. 2.4, is embedded in the generalized MIDO formulation for the design optimization problem[2]. The exact implementation of

[2]The scheduling actions are assumed to be constant in Eq. 1.11 for this particular class of problem, as previously

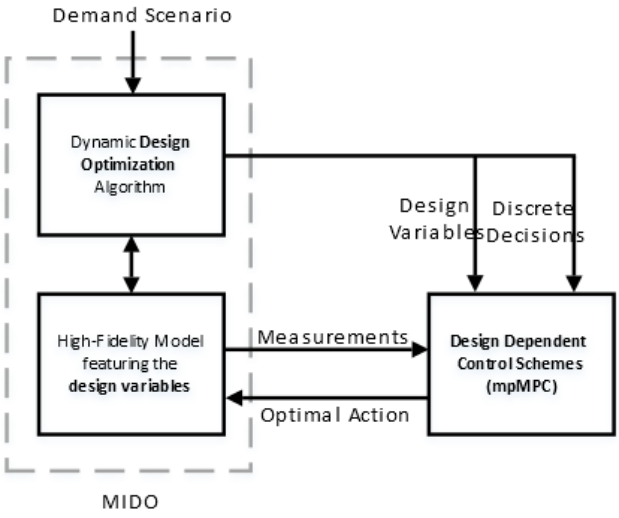


Figure 2.2: Schematic representation of the simultaneous design MIDO with embedded mpMPC. The area within the dashed line represents the MIDO problem (Reprinted with permission from [88]).

the control strategy allows for the simultaneous consideration of the design and control problems. The flow of information to solve the resulting MIDO problem is illustrated in Figure 2.2.

The following section showcases the described framework on a binary distillation column.

2.3 Case Study – A binary distillation column

The distillation column model describes a binary separation process of benzene and toluene. The column is allowed a maximum number of trays to be 30 with no restriction on feed tray location. The purity in the top has a desired set point of 0.98 and the purity in the bottom has a desired setpoint of 0.02. The feed composition is assumed sinusoidal.

2.3.1 High-Fidelity dynamic modeling

The distillation column utilizes mass and energy balances and thermodynamic relations to build the full model. It has been assumed that (i) the energy dynamics are fast compared to mass transfer dynamics, (ii) relative volatility is constant throughout the column, (iii) molar holdup is constant in the condenser, (iv) the system responds immediately to the changes in pressure.

mentioned in Section 2.1.

Table 2.2: Characteristic equations for a binary distillation column. Component i is benzene, tray number is $k \in \{1...Ntrays\}$ unless stated otherwise (Reprinted with permission from [88]).

Description	Equation
Component mass balance	$\frac{dM_{i,k}}{dt} = L_{k+1}x_{i,k+1} + V_{k-1}y_{i,k-1} + F_k z_{i,f} + R_k x_{i,D} - L_k x_{i,k} - V_k y_{i,k},\ \forall k \in \{2...Ntrays-1\}$
Total mass balance	$\frac{dM_k}{dt} = L_{k+1} + V_{k-1} + F_k + R_k - L_k - V_k,\ \forall k \in \{2...Ntrays-1\}$
Vapor molar flow rate	$V_k = V_{k-1} = V_B,\ \forall k \in \{2...Ntrays-1\}$
Hold-up	$Vol_k = \frac{M_k}{\rho_{Lmix,k}}$
Liquid level	$Level_k = \frac{L_k^{2/3}}{1.84\rho_{Lmix,k}L_{weir}} + H_{weir}$
Tray area	$A_{tray} = \frac{0.8\pi D_c^2}{4}$
Weir length	$L_{weir} = 0.77D_c$
Reboiler vapor liquid equilibrium	$1 = \frac{P^0_{benz,B}x_{i,B} + P^0_{tol,B}(1-x_{i,B})}{P}$
Condenser vapor liquid equilibrium	$1 = P(\frac{x_{i,D}}{P^0_{benz,D}} + \frac{1-x_{i,D}}{P^0_{tol,D}})$
Relative volatility	$\alpha = \sqrt{\frac{P^0_{benz,D}P^0_{benz,B}}{P^0_{tol,D}}P^0_{tol,B}}$
Reboiler and reflux drum molar balance	$y_{i,k} = \frac{\alpha x_{i,k}}{1+x_{i,k}(\alpha-1)}$ $\frac{dM_{i,B}}{dt} = L_1x_{i,1} - Bx_{i,B} - V_B y_{i,B}$ $M_B = \frac{M_{i,B}}{x_{i,B}}$ $\frac{dM_{i,D}}{dt} = V_{Ntrays}(y_{i,Ntrays} - x_{i,D})$ $M_D = \frac{M_{i,D}}{x_{i,D}}$
Reboiler and reflux drum energy balance	$0 = L_1 - B - V_B$ $0 = V_D - \Sigma R_k - D$

The complete model of the binary distillation column is adapted from Sakizlis et al., (2003) [86], and the characteristic equations of the model are presented in Table 2.2.

Mass balances for each tray, reboiler, and condenser are used while assuming constant molar hold up in the total condenser. Energy balances are used in the reboiler and condenser assuming an average temperature throughout the column. Relative volatility is used to determine vapor and liquid correlations in each tray and in the reboiler. The model assumes the reflux flow rate and the boilup rate to be the controllable variables in the system, and the molar hold ups to be the states of the system. Column diameter, reflux tray position, and feed tray position are the design variables, while the presence and position of the reboiler and condenser are fixed. Density of the liquid hold

up on the trays is assumed to follow from a linear combination of the component densities.

Antoine equations were used to determine vapor pressures at the top and bottom of the distillation column and the log-mean temperature approach was used for the heat exchange at the condenser and the reboiler. The distillation column is a multiple input multiple output (MIMO) system where the reflux flow rate and vapor flow rate are the degrees of freedom to the system, and the purity in the top and bottom is the output. The composition at the feed is treated as a disturbance to the system operation.

2.3.2 Model approximation

The high fidelity model of the distillation column consists of 52 states and nonlinear equations. Random sets of input/output data for different designs from the high fidelity model are introduced into the System Identification Toolbox in MATLAB® to acquire a linear state-space model of the form of Eq. 2.2. The identified state-space model is shown in Eq. 2.5.

$$
\begin{aligned}
x_{t_c+1} = &\begin{bmatrix} 0.9533 & -0.05507 \\ 0.0264 & 0.5494 \end{bmatrix} x_{t_c} + \begin{bmatrix} -0.01609 & -0.01346 \\ -0.1129 & 0.08987 \end{bmatrix} u_{t_c} + \\
&\begin{bmatrix} -0.1257 & -9.703 \cdot 10^{-5} & -4.163 \cdot 10^{-4} \\ -1.005 & 7.184 \cdot 10^{-4} & -5.874 \cdot 10^{-5} \end{bmatrix} \begin{bmatrix} d_{t_c} \\ des \end{bmatrix} \\
\hat{y}_{t_c} = &\begin{bmatrix} -0.2357 & -0.354 \\ 0.1098 & -0.4719 \end{bmatrix} x_{t_c} \\
t_c = &\ 1s
\end{aligned}
\tag{2.5}
$$

where x_{t_c} are the identified states, u_{t_c} are the reflux flow rate and the vapor rate, d_{t_c} is the composition of the feed and des is the feed and reflux tray location. The performance of the approximate model to capture the dynamics of the high-fidelity model is presented by Figures 2.3 and 2.4, where the step and impulse responses of the approximate model is demonstrated. Note that the column diameter is correlated to the minimum vapor flow rate and therefore is the design decision of the system (See Eq. 2.6). The number of trays, and the location of the feed tray are

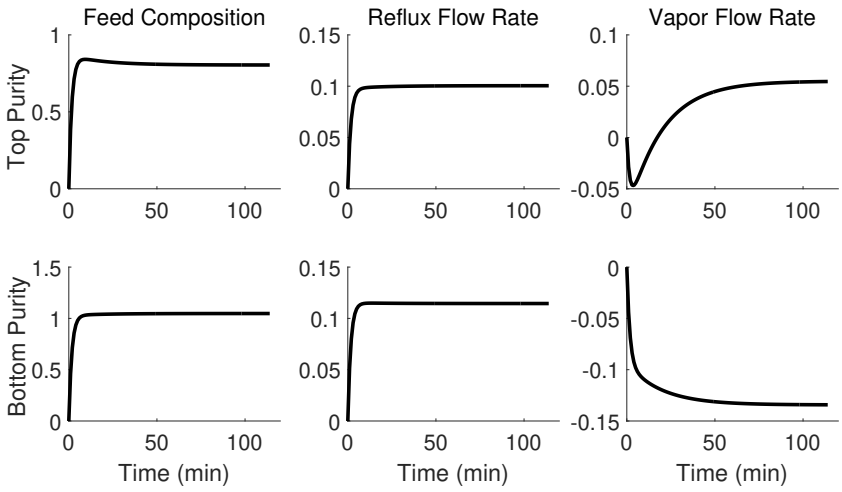


Figure 2.3: Step response of the approximate model for the binary distillation column example (Reprinted with permission from [88]).

also accounted for as design decisions, however addressing these integer variables is discussed in the following subsection.

2.3.3 Design dependent mpMPC

The problem formulation of the control strategy is based on Eq. 2.3 and the tuning of the controller is presented in Table 2.3. Note that since the vapor flow rate is limited by the column diameter as presented in Eq. 2.6, the mpMPC is modified to account for the square of the column diameter as a design parameter. Note that since the column diameter is always greater than zero and it does not appear anywhere else within the mpMPC formulation we can define a new parameter $p = D_c^2$ which renders Eq. 2.6 a linear inequality constraint.

$$0.4514 \cdot V_B \leq D_c^2 \tag{2.6}$$

The integer parameters corresponding to the tray locations are reformulated into binary parameters and solved based on the algorithm presented in [162]. An alternative formulation could be the treatment of integer parameters as continuous parameters (similar to handling binary variables in [163]) since the integer value realization of the parameter is a subset of their continuous values

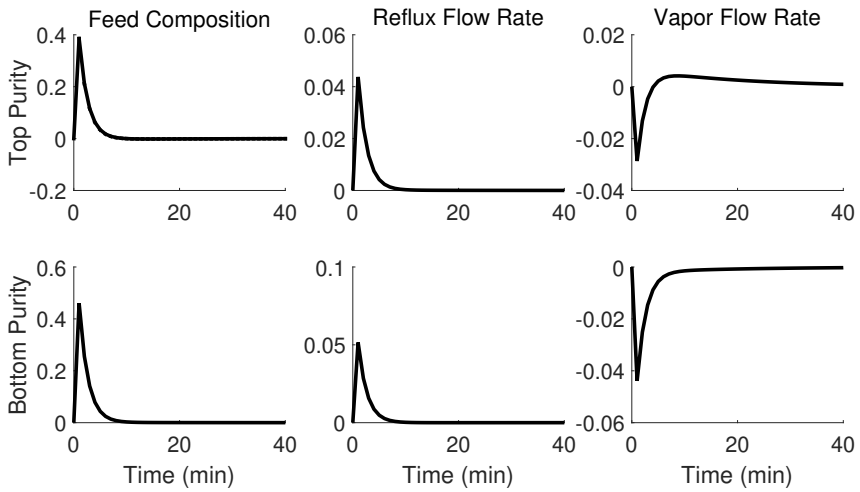


Figure 2.4: Impulse response of the approximate model for the binary distillation column example (Reprinted with permission from [88]).

Table 2.3: Weight tuning for the mpMPC of the distillation column (Reprinted with permission from [88]).

MPC design parameters	Value
N_c	3
M_c	1
$QR_k, \forall k \in \{1, ..., \mathbf{N}\}$	$\begin{bmatrix} 10^7 & 0 \\ 0 & 10^7 \end{bmatrix}$
$R_k, \forall k \in \{1, ..., \mathbf{M}\}$	$\begin{bmatrix} 10^{-2} & 0 \\ 0 & 10^{-2} \end{bmatrix}$
$\underline{x}$	$\begin{bmatrix} -10^3 & -10^3 \end{bmatrix}^T$
$\overline{x}$	$\begin{bmatrix} 10^3 & 10^3 \end{bmatrix}^T$
$\underline{u}$	$\begin{bmatrix} 2 & 3 \end{bmatrix}^T$
$\overline{u}$	$\begin{bmatrix} 4.7 & 7 \end{bmatrix}^T$
$\underline{y}$	$\begin{bmatrix} 0 & 0 \end{bmatrix}^T$
$\overline{y}$	$\begin{bmatrix} 1 & 1 \end{bmatrix}^T$
$\underline{d}$	$\begin{bmatrix} 0.45 & 1 & 1 \end{bmatrix}^T$
$\overline{d}$	$\begin{bmatrix} 0.5 & 30 & 30 \end{bmatrix}^T$

and the realization is not an mpMPC decision.

The objective of the design dependent controller is to maintain the purity set points for the top and bottom product at 98% vol and 2% vol regardless of the disturbance at the inlet of the system. Deviations from the desired set points are allowed but penalized in the design optimization formulation.

2.3.4 Closed loop validation

The developed control strategy is validated against the high-fidelity model, under a range of design realizations. Figure 2.5 shows the closed loop profile for an arbitrary design configuration. It is observed that the controller is (i) able to track a set point, (ii) satisfies the path constraints, and (iii) maintains the stability of the system under external process disturbances.

2.3.5 Dynamic optimization

The dynamic optimization is formulated with the explicit map of optimal control strategy and solved allowing for the optimizer to select the optimal value for the area of the condenser, area of the reboiler, reflux tray location, feed tray location, and diameter of the column. To account for the reflux and feed tray location changing additional equations were added or modified as seen in Table 2.4.

Allowing the dynamic optimization to run over a time span of 1 hour, the results obtained are presented in Table 2.5. It can be seen that by utilizing the simultaneous design and control approach presented here, a distillation column with a smaller annualized total cost is designed.

2.4 Conclusions

In this chapter, an integrated framework was presented for the application of process design and control optimization via multiparametric programming. Explicit expressions were developed for the design dependent optimal control actions, which allowed for using a single offline control strategy to be employed under a range of design configurations. The design and control problems were then simultaneously solved by incorporating the map of design dependent control strategy in the design optimization problem and the solving the integrated MIDO. The presented framework

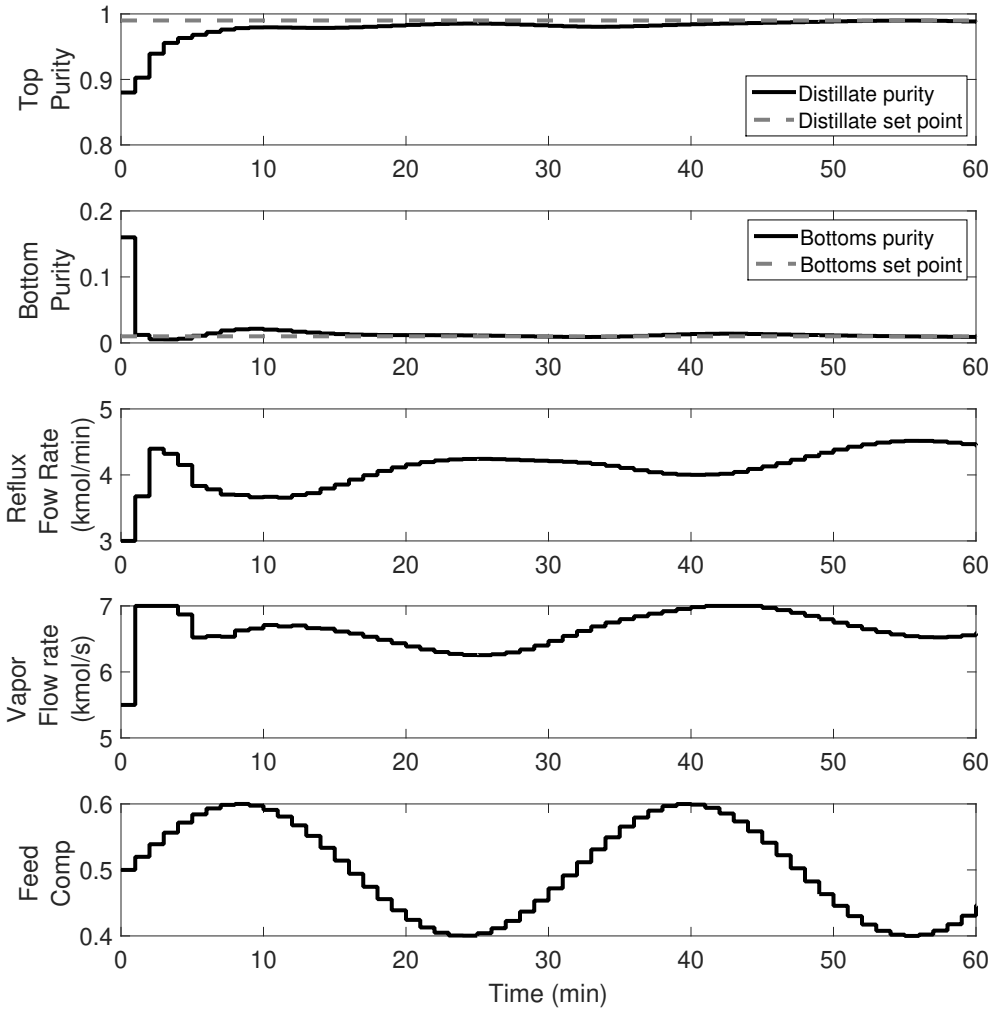


Figure 2.5: Closed loop validation of the control strategy against the high-fidelity model. The arbitrarily fixed design variables are as follows. Diameter of the column: $1.9m$, Reflux tray position: 18, Feed tray position: 9 (Reprinted with permission from [88]).

Table 2.4: Additional/Modified equations for dynamic optimization. Component i is benzene, tray number is $k \in \{1...Ntrays\}$ unless stated otherwise (Reprinted with permission from [88]).

Description	Equation
Feed tray location	$F_k = F\delta_k^f,\ \sum_{k=1}^{Ntrays} \delta_k^f = 1$
Reflux tray location	$R_k = R\delta_k^r,\ \sum_{k=1}^{Ntrays} \delta_k^r = 1$
Feed tray location only below reflux	$\delta_k^f - \sum_{k'=k}^{Ntrays} \delta_{k'}^r \leq 0$
Component mass balance	$(\sum_{k'=k}^{Ntrays} \delta_{k'}^r)\frac{dM_{i,k}}{dt} = L_{k+1}x_{i,k+1} + V_{k-1}y_{i,k-1} + F_k z_{i,f} + R_k x_{i,d} - L_k x_{i,k} - V_k y_{i,k},\ \forall k \in \{2.., Ntrays - 1\}$
Total mass balance	$(\sum_{k'=k}^{Ntrays} \delta_{k'}^r)\frac{dM_k}{dt} = L_{k+1} + V_{k-1} + F_k + R_k - L_k - V_k,\ \forall k \in \{2.., Ntrays - 1\}$
Reboiler cost	$C_{reb} = 0.6 \cdot 101.3\frac{M\&S}{280}(\frac{10^4 A_R}{144 \cdot 2.54^2})^{0.65} \cdot 3.22 \cdot 1.35$
Total cost	$TotalCost = OpCost + \frac{1}{3}(C_{column} + C_{reb} + C_{cond})$

Table 2.5: Results of the current approach and comparison against Sakizlis et al. (2003) [86] (Reprinted with permission from [88]).

	Current Approach	Comparison against Sakizlis et al. [86]
Condenser Area [m^2]	120	132
Reboiler Area [m^2]	266	276
Diameter of Column [m]	1.62	1.65
Reflux Tray	25	25
Feed Tray	12	12
Total Cost [k$]	590	620

was showcased on a binary distillation column, and a comparison was made with a previously introduced integration methodology. The comparison revealed that the recently proposed framework yielded lower operating and capital costs.

Although the presented framework was shown to be effective to integrate the design and control problems, its applicability is limited with the operations where the operating region is limited and known prior to the realization of the process. For instance in the binary distillation column example, we knew a priori that the product purities would be around 98%. However, the optimal operating region is typically an economical decision that needs to account for the state of the system along with the market conditions, including the raw material and utility costs, demand on the products, availability of the process equipment. Therefore in the next chapter, the proposed framework will be modified to account for the scheduling level decisions.

3. INTEGRATION OF PROCESS SCHEDULING AND CONTROL[1]

3.1 Introduction

Production scheduling and process control are two layers in the process operations that are highly dependent due to the volume of reciprocal information flow. Typically, the process schedule coordinates the production sequence, production times, and inventory levels based on the market dynamics. Process control, on the other hand, delivers the production targets with the existence of operational uncertainty, measured/unmeasured process disturbances, and plant-model mismatch. These layers are typically addressed independently and sequentially due to the hierarchical nature of the underlying problems. The isolation between the decisions from different layers can result in suboptimal, or even infeasible operations [164, 94].

Individual assessment of the scheduling and control problems requires assumptions that neglect the dynamics introduced by their complements. The scheduling problem utilizes static tables comprising the process time constants for the transitions between the operating modes of the system. These time constants are typically obtained by exhaustive closed-loop simulations conducted offline. Consequently, the static tables fail to represent the closed-loop dynamics of the system due to the lack of an underlying high-fidelity model [103, 97].

A simultaneous approach for process scheduling and control reconstructs the two problems as a unified problem. The reformulated problem takes into account the degrees of freedom of the two subproblems simultaneously, leading to an augmented feasible space. This allows the chemical plant to respond to rapidly changing market conditions while maintaining feasible and profitable operation. These changes include but are not limited to the market demand, price, and the spectrum and specifications regarding the products manufactured in the chemical plant. Furthermore, fluctuating operating costs require flexibility in the process scheduling [94]. Therefore, a chemical

[1]Portions of this chapter have been adapted with permission from Burnak, B., Katz, J., Diangelakis, N.A., Pistikopoulos, E.N., Simultaneous Process Scheduling and Control: A Multiparametric Programming-Based Approach, Industrial & Engineering Chemistry Research 2018, 57, pp. 3963-3976. Copyright (2018) American Chemical Society.

Table 3.1: Scheduling and control in the literature: An indicative list (Reprinted with permission from [105]).

Author (year)	Contribution
Grossmann & co-workers (2006, 2007, 2010, 2011, 2012, 2014) [97, 98, 99, 100, 166, 167], Gudi & co-workers (2010) [168], Biegler & co-workers (2012, 2014) [169, 170], Espuña & co-workers (2013) [171], You & co-workers (2013) [172], Baldea & co-workers (2016, 2018) [173, 174]	Decomposition of MIDO or MINLP and open loop optimal control
Allcock & co-workers (2002) [95], Pistikopoulos & co-workers (2003) [96], Ierapetritou & co-workers (2012) [101], You & co-workers (2012) [175], Baldea & co-workers (2015, 2018) [103, 176]	Formulation/ Decomposition of MIDO schedule with PID control
Ierapetritou & co-workers (2014, 2018) [177, 178], Christofides & co-workers (2014a, 2014b, 2015, 2017) [112, 179, 180, 181], Baldea & co-workers (2015, 2018) [104, 178], Swartz & co-workers (2017) [113], Pistikopoulos & co-workers (2017, 2018) [131, 105], Dua & co-workers (2019) [106], Hedengren & co-workers (2018) [182]	(mp)MPC implementation in economic receding horizon policies
Puigjaner & co-workers (1995) [183], Marquardt & co-workers (2011) [111], Rawlings & co-workers (2011, 2012, 2013) [110, 184, 185], Pistikopoulos & co-workers (2013, 2014) [186, 187], Baldea & co-workers (2014) [188], Liu & co-workers (2016) [189]	Control theory/ Economic MPC in scheduling problems
Reklaitis & co-workers (1996) [190], Grossmann (2005) [1], Harjunkoski & co-workers (2009) [165], Engell & Harjunkoski (2012) [164], Baldea & co-workers (2014) [94], Christofides & co-workers (2014) [191], You & co-workers (2015) [192], Ierapetritou & co-workers (2016, 2017) [193, 194]	Review articles on scheduling and control integration

process needs integrated decisions that enable higher adaptability and operability to remain competitive in the market [165]. There have been some attempts over the years to tackle the two aspects of operational optimization in an integrated framework. An indicative list of these contributions is presented in Table 3.1.

Over two decades of academic literature on integrated approaches for the process scheduling and control problem has focused on a systematic methodology to overcome the following fundamental challenges [104]:

(i) *Discrepancies in objectives*: The schedule and control formulations are designed to deliver

specialized tasks in a process. The former aims to bring profitable operation by taking into account the operational aspects such as the process economics, raw material and equipment availability, and product specifications; while the latter involves real-time manipulation of select process variables to meet the targeted product specifications. These scheduling and control goals are not always aligned and frequently require the compromise of one of their respective objectives.

(ii) *Discrepancies in time-scales*: A typical control horizon varies between seconds and minutes, whereas the scheduling horizon is on the order of hours or weeks. Therefore, integration of the two distinct problems into a unified formulation creates a large scale, stiff system due to the order of magnitude differences in their respective time scales [104, 103]. Following direct solution approaches for the reformulated unified problem have been shown to be computationally intractable [170].

In this chapter, a surrogate model formulation is presented to bridge the inherent gap between the schedule and control formulations. The surrogate model is designed to translate the fast closed-loop dynamics to the slower scheduling dynamics, while providing corrective time varying targets for the controller. We utilize the reactive scheduling approach introduced by Subramanian et al. [184], and adapted in a multiparametric framework by Kopanos and Pistikopoulos [187], formulating a state-space representation that is implemented in a rolling horizon framework. This formulation is solved once and offline via multiparametric programming techniques, deriving optimal scheduling decisions as affine functions of the product demand scenarios. The derivation of the controllers, on the other hand, is adapted from the PAROC framework [130], which provides a systematic methodology to design advanced model-based controllers via multiparametric programming.

The remainder of the chapter is organized as follows. In Section 3.2 derivation of a control-aware scheduler, and a schedule-aware controller using the PAROC framework is introduced. In Section 3.3, a CSTR example with a three raw materials and three products is presented to showcase the framework, and the example is generalized to multiple CSTRs operating in parallel. Fi-

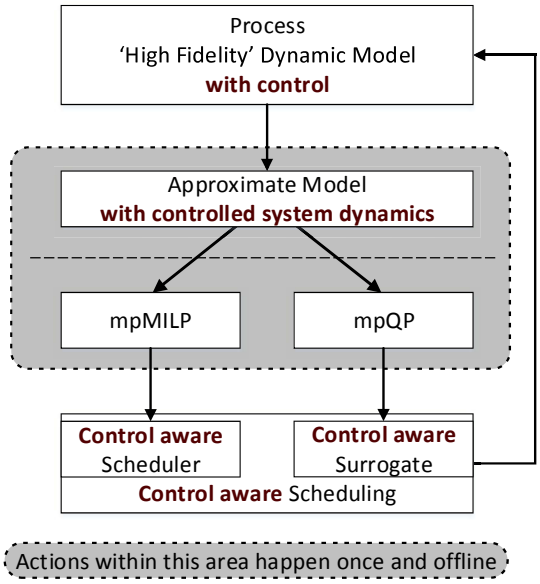


Figure 3.1: Derivation of the scheduling scheme. Actions within the gray area happen once and offline (Reprinted with permission from [105]).

nally, the conclusions are summarized in Section 3.4.

3.2 Simultaneous scheduling and control via PAROC

The objective and the function of the PAROC framework and software prototype has been previously introduced in Chapter 2, where the integrated design and control optimization has been addressed. Here, we demonstrate the applicability of the framework on the simultaneous scheduling and control problem.

First, we derive an offline schedule-aware control strategy that is applicable under distinctly separate range of process conditions. This offline strategy is embedded into the original mathematical model, and a control-aware approximate model is derived to describe the closed-loop behavior of the system. The resulting model is used to generate offline maps of (i) long term decisions regarding the operational feasibility and profitability and (ii) a surrogate model to bridge the gap between the short term and long term decisions. The offline maps are validated against the high-fidelity model used in the first step. A schematic representation of the proposed methodology is presented in Figure 3.1 [105]. The closed-loop implementation of the framework and the fundamental interactions between different layers of models for the integration of schedule and control

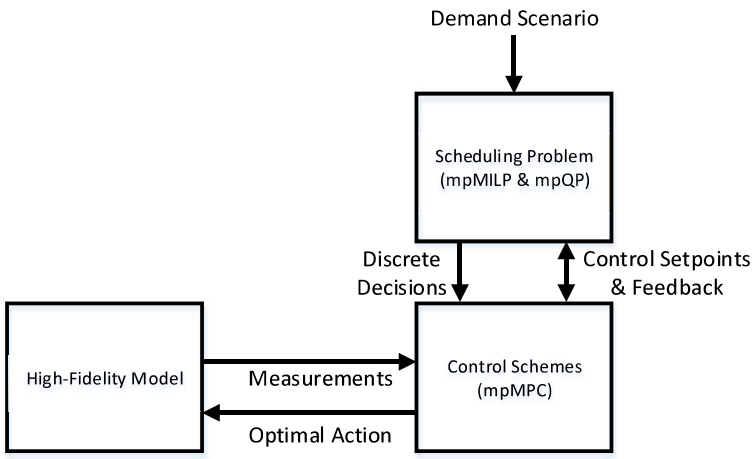


Figure 3.2: Schematic representation of the simultaneous scheduling and control (Reprinted with permission from [105]).

are depicted in Figure 3.2 [105].

Note the following advantages of solving the multiparametric counterparts of the schedule, control, and time scale bridging surrogate model:

- Offline maps of optimal operations at both long and short terms are acquired as explicit expressions.
- Online computational time for the optimal control problem is reduced to a simple look-up table algorithm and evaluation of an affine function. Such significant reduction enables the application of the framework to system with fast dynamics.
- The offline maps of solutions can allow for the integration of the design of the process/equipment with the schedule and control in a dynamic optimization framework. We will see the benefits of this advantage in the next chapter, where we formulate an integrated design, scheduling, and control problem utilizing the generated offline rolling horizon operating strategies.

Following are the fundamental steps of PAROC in further detail, tailored specific to the needs of the simultaneous scheduling and control problem.

3.2.1 Designing schedule-aware controller

Developing the offline control strategy is similar to the procedure described in Section 2.2, hence only the key differences are highlighted in this section.

Step 1 – High-fidelity dynamic modeling: Similar to the integrated design and control problem, we start with constructing a rigorous mathematical model that simulates the dynamics of the system of interest. The general representation of the high fidelity model is given by Eq. 3.1.

$$\begin{aligned} \dot{x}(t) &= f(x(t), u(t), s(t), d(t), t) \\ y(t) &= g(x(t), u(t), s(t), d(t), t) \end{aligned} \tag{3.1}$$

where x are the states, u are the control decisions, s are the scheduling decisions, d are the external disturbances, t is independent time variable, y are the system outputs, and f and g are generic functions.

Step 2 – Model approximation: Model approximation is performed via the MATLAB® System Identification Toolbox™ yielding the state space representation in Equation 3.2.

$$\begin{aligned} x^q_{t_c+1} &= A^q x^q_{t_c} + B^q u_{t_c} + C^q [d^T_{t_c}, Sc^T_{t_c}]^T \\ \hat{y}_{t_c} &= D^q x^q_{t_c} + E^q u_{t_c} + F^q [d^T_{t_c}, Sc^T_{t_c}]^T \end{aligned} \tag{3.2}$$

where A, B, C, D, E, and F are matrices with appropriate dimensions, superscript q denotes the index of the linear model, Sc denotes the degrees of the system that is determined by the scheduler and unavailable to the controller, $\hat{y}$ are the outputs predicted by the controller, and t_c is the discretization step of the state space model. Note that the states $x^q_{t_c}$ can be concatenated into a single vector, x_{t_c}.

Remark 2. *Due to the nature of the problem, the controller is expected to operate at multiple regions. Therefore, a single state space model may be inadequate to capture the entire process dynamics. This problem is addressed by deriving multiple state space models that will be utilized in nonoverlapping operating regions. The formulation to define the domain of each state space model is achieved by mixed-integer modeling in designing the control strategy, which will be*

demonstrated in the example presented in Section 3.3.1.

Step 3 – Multiparametric model predictive control (mpMPC): The control scheme is designed based on the developed approximate model given by Eq. 3.2. Note that the mpMPC is aware of the scheduling level decisions since the state space model incorporates them as measured disturbances. Equation 3.3 describes the general form of the mpMPC formulation used in this chapter.

$$
\begin{aligned}
u_{t_c}(\theta) = \arg\min_{u_{t_c}} \quad & \|x_{N_c}\|_P^2 + \sum_{t_c=1}^{N_c-1} \|x_{t_c}\|_Q^2 + \sum_{t_c=1}^{N_c-1} \|y_{t_c} - y_{t_c}^{SP}\|_{QR}^2 \\
& + \sum_{t_c=0}^{M_c-1} \|u_{t_c} - u_{t_c}^{SP}\|_R^2 + \sum_{t_c=0}^{M_c-1} \|\Delta u_{t_c}\|_{R1}^2 \\
s.t. \quad & x_{t_c+1} = Ax_{t_c} + Bu_{t_c} + C[d_{t_c}^T, Sc_{t_c}^T]^T \\
& \hat{y}_{t_c} = Dx_{t_c} + Eu_{t_c} + F[d_{t_c}^T, Sc_{t_c}^T]^T \\
& y_{t_c} = \hat{y}_{t_c} + e \\
& e = y_{t_c=0} - \hat{y}_{t_c=0} \\
& \underline{x} \leq x_{t_c} \leq \overline{x}, \quad \underline{y} \leq y_{t_c} \leq \overline{y} \\
& \underline{u} \leq u_{t_c} \leq \overline{u}, \quad \underline{\Delta u} \leq \Delta u_{t_c} \leq \overline{\Delta u} \\
& \theta = [x_{t_c=0}^T, u_{t_c=-1}^T, d_{t_c=0}^T, s_{t_c}^T]^T \\
& \{y_{t_c}^{SP}, u_{t_c}^{SP}, Sc_{t_c}\} \subseteq s_{t_c}, \ \forall t_c \in \{0, 1, ..., N_c - 1\}
\end{aligned}
\tag{3.3}
$$

where x are the state variables, y are the system outputs, u are the control variables, Δu are the magnitude between two consecutive control actions, d are measured disturbances, Q, QR, R, $R1$ are the corresponding weights in the objective function, P is the stabilizing term determined by solving the discrete time algebraic Ricatti equation, N and M are the output horizon and control horizon, respectively. e denotes the plant-model mismatch and is defined as the difference between the real output measured and the state space estimation of the output at $t_c = 0$. y^{SP} and u^{SP} are the output set points and input reference points, respectively. Note that these two vectors of variables are scheduling level decisions (i.e. $\{y^{SP}, u^{SP}\} \subseteq s$), and hence mpMPC treats them as additional

parameters in θ.

The minimization problem presented in Equation 3.3 is translated into a linearly constrained quadratic multiparametric programming problem (mpQP) via the YALMIP toolbox [195], and solved via the Parametric OPtimization (POP) Toolbox [161] in MATLAB®. The solution of the mpQP problem yields explicit control actions as an affine function of the uncertain parameters, as presented in Equation 3.4.

$$\begin{aligned} & u_j(\theta) = K_n\theta + r_n, \forall\theta \in CR_n \\ & \theta := \left[x^T_{t_c=0}, u^T_{t_c=-1}, d^T_{t_c=0}, Sc^T_{t_c}, (y^{SP}_{t_c})^T, (u^{SP}_{t_c})^T, y^T_{t_c=0}\right]^T \\ & CR_n := \{\theta \in \Theta \mid L_n\theta \leq b_n\}, \forall n \in \{1, 2, ...NC\} \\ & \forall j \in \{0, 1, ..., M_c\}, \forall t_c \in \{0, 1, ..., N_c\} \end{aligned} \tag{3.4}$$

where θ is the set of uncertain parameters measured at $t_c = 0$, $u_{t_c=-1}$ is the optimal control action at the previous time step, CR_n is the active polyhedral partition of the feasible parameter space, NC is the number of critical regions CR_n, and Θ is defined as a closed and bounded set. Note that inclusion of scheduling level decisions, i.e. Sc_{t_c}, $y^{SP}_{t_c}$, and $u^{SP}_{t_c}$ in the parameter space enables mpMPC to account for any future changes in the operational level *a priori* within the range of the output horizon.

Step 4 – Closed-loop validation: Since the framework suggests an approximation of the high fidelity model, a validation step is mandatory to test the validity of the simplified model, as well as the controller scheme. Therefore, the mpMPC derived in Step 3 is validated through in-silico testing against the high fidelity model in Step 1.

3.2.2 Designing control-aware scheduler

Production scheduling of a chemical process formulated as a general MILP problem can also be represented by a state space model [184, 185]. Multiparametric counterpart of this class of reactive scheduling problems and its solution is described extensively in Kopanos and Pistikopoulos [187]. This approach yields an optimal map of solutions under potential disruptions in the course

of operation prior to the occurrence of the event. The explicit form of the schedule significantly reduces the computational cost of repetitive evaluations after every disruptive event. However, the sampling time of the state space model is typically too large to account for the dynamic considerations inherent to the process. Hence, such an approach suggests utilization of static transition tables based on exhaustive testing [103] that create plant-model mismatch since they are agnostic to the real system dynamics.

In this chapter, a two level scheduling scheme with a hierarchical order is proposed: (i) an upper level schedule for the regulation of the economic considerations and operational feasibility based on the formulation of Kopanos and Pistikopoulos, (2014) [187], and (ii) a lower level surrogate model to bridge the time scales between the control and the upper level schedule based on the closed loop behavior of the high fidelity model. The surrogate model further aims to remedy the plant-model mismatch introduced by the schedule.

Step 1 – High fidelity model with controller embedded: The control scheme derived in the earlier phase (Eq. 3.4) is embedded in the original high fidelity model (Eq. 3.1).

Step 2 – Approximate models: A discrete time state space model is derived based on the closed loop behavior of the high fidelity model. The input-output relationship focuses on capturing the overall response of the closed loop system to the step changes in the output set points and input reference trajectories. Note that the discretization time of the identified model for the upper level scheduler is several orders of magnitude larger than the mpMPC. Therefore, we introduce a surrogate model formulation to translate the upper level scheduling decisions in the first scheduling time step into the control time steps. This translation is carried out by resampling the identified scheduling model with a discretization step matching the output horizon of the mpMPC. The resampled model is used as the governing constraint in the surrogate model formulation, as described in detail in the next step.

Step 3 – Design of the multiparametric schedule and surrogate model: The multiparametric schedule is formulated with an objective to account for the economic considerations and operational feasibility, subjected to the corresponding approximate model derived earlier, as de-

scribed in Kopanos and Pistikopoulos, (2014) [187]. The resulting formulation creates a mpMILP that treats the disruptive scheduling events as parameters described in Eq. 3.5.

$$
\begin{aligned}
\underset{s_{t_s}}{min} \quad & J(\theta) = \sum_{t_s=1}^{N_s} \alpha^T \tilde{x}_{t_s} + \sum_{t_s=0}^{N_s-1} \beta^T \tilde{tr}_{t_s} + \sum_{t_s=0}^{N_s} \phi^T \tilde{u}_{t_s} \\
s.t. \quad & \tilde{x}_{t_s+1} = A_1 \tilde{x}_{t_s} + B_1 \tilde{u}_{t_s} + C_1 \tilde{d}_{t_s} \\
& \tilde{tr}_{t_s} = A_2(\tilde{x}_{t_s} - \tilde{x}_{t_s-1}) + B_2(\tilde{u}_{t_s} - \tilde{u}_{t_s-1}) \\
& \tilde{u}_{t_s} = \left[\tilde{Sc}_{t_s}^T, (\tilde{y}_{t_s}^{SP})^T, (\tilde{u}_{t_s}^{SP})^T\right]^T \\
& \theta = \left[\tilde{x}_{t_s=0}^T, \tilde{x}_{t_s=-1}^T, \tilde{u}_{t_s=-1}^T, \tilde{d}_{t_s}^T\right]^T \\
& \underline{x}_{t_s} \leq \tilde{x}_{t_s} \leq \overline{x}_{t_s} \\
& \underline{tr}_{t_s} \leq \tilde{tr}_{t_s} \leq \overline{tr}_{t_s} \\
& \underline{u}_{t_s} Y_{t_s} \leq \tilde{u}_{t_s} \leq \overline{u}_{t_s} Y_{t_s} \\
& s_{t_s} = [\tilde{u}_{t_s}, Y_{t_s}], \quad \forall t_s \in \{0, 1, ..., N_s\}
\end{aligned}
\tag{3.5}
$$

where the tilde ($\sim$) sign denotes a scheduling level counterpart of the variable, $\tilde{x}$ is the operational level and the inventory, $\tilde{tr}$ denotes transition to a different operational mode, and the Greek letters α, β, and ϕ are the corresponding cost parameters. Note that additional constraints can be included in Equation 3.5 regarding the needs of the specific problem. The multiparametric solution of Equation 3.5 provides explicit affine expressions of the optimal scheduling actions as functions of the system parameters, as defined in Equation 3.6.

$$
\begin{aligned}
& s_{t_s} = \tilde{K}_n \theta + \tilde{r}_n, \forall \theta \in CR_n \\
& \theta := \left[\tilde{x}_{t_s=0}^T, \tilde{x}_{t_s=-1}^T, \tilde{u}_{t_s=-1}^T, \tilde{d}_{t_s}^T\right]^T \\
& CR_n := \{\theta \in \Theta \mid \tilde{L}_n \theta \leq \tilde{b}_n\}, \forall n \in \{1, 2, ...NC\} \\
& \forall t_s \in \{0, 1, ..., N_s\}
\end{aligned}
\tag{3.6}
$$

Due to approximation of the scheduling model and the large discretization time, there exists a plant-model mismatch that is handled by a surrogate model formulated as a mp(MI)QP. Therefore,

we utilize the formulation presented in Equation 3.7 to minimize the aforementioned mismatch.

$$
\begin{aligned}
\underset{s_{t_{sm}}}{min} \quad & J(\theta) = \sum_{t_{sm}=0}^{N_{sm}} \|s_{t_{sm}} - \tilde{u}_{t_s}\|^2_{R'} \\
s.t. \quad & x'_{t_{sm}+1} = Ax'_{t_{sm}} + Bs_{t_{sm}} \\
& y_{t_{sm}} = Cx'_{t_{sm}} + Ds_{t_{sm}} \\
& s_{t_{sm}} = \left[Sc^T_{t_{sm}}, (y^{SP}_{t_{sm}})^T, (u^{SP}_{t_{sm}})^T\right]^T \\
& \theta = \left[\tilde{u}^T_{t_{sm}}, Y^T_{t_{sm}}\right] \\
& \underline{x}'_{t_{sm}} \leq x_{t_{sm}} \leq \overline{x}'_{t_{sm}} \\
& \underline{y}_{t_{sm}} \leq y_{t_{sm}} \leq \overline{y}_{t_{sm}} \\
& \underline{s}_{t_{sm}} Y_{t_{sm}} \leq s_{t_{sm}} \leq \overline{s}_{t_{sm}} Y_{t_{sm}} \\
& \forall t_{sm} \in \{0, 1, ..., N_{sm}\}
\end{aligned}
\tag{3.7}
$$

Equation 3.7 poses an mpQP problem that reinterprets the scheduling actions s_{t_s} in the time steps of the controller. $Sc_{t_{sm}}$ is directly passed to the process, and the set points $y^{SP}_{t_{sm}}$ and $u^{SP}_{t_{sm}}$ are determined to be used by the controller. Δt_{sm} is based on the output horizon of the mpMPC ($\Delta t_{sm} = \Delta t_c N_c$), and N_{sm} is selected such that the surrogate model horizon can account for the first scheduling time step in its entirety (i.e. $N_{sm} \geq \Delta t_s / \Delta t_{sm}$). The multiparametric solution to Eq. 3.7 yields an offline map of optimal scheduling actions and set points for the controller, allowing for fast reevaluation of the scheduling decisions under varying market conditions. The surrogate model formulation utilizes a linear state space representation of the closed loop dynamics of the system. Therefore, the number of state space models required to capture the complete dynamics is dependent on the complexity of the high fidelity model and the size of the explicit control law. The validity of these surrogate models representations is assured in the subsequent step.

Note that binary decisions $Y_{t_{sm}}$ from Eq. 3.5 are treated as continuous uncertain parameters. Oberdieck *et al.* (2016) [162] presents a rigorous proof through Basic Sensitivity Theorem that relaxation of the binary parameters yields the exact solution in this class of problems. Equiva-

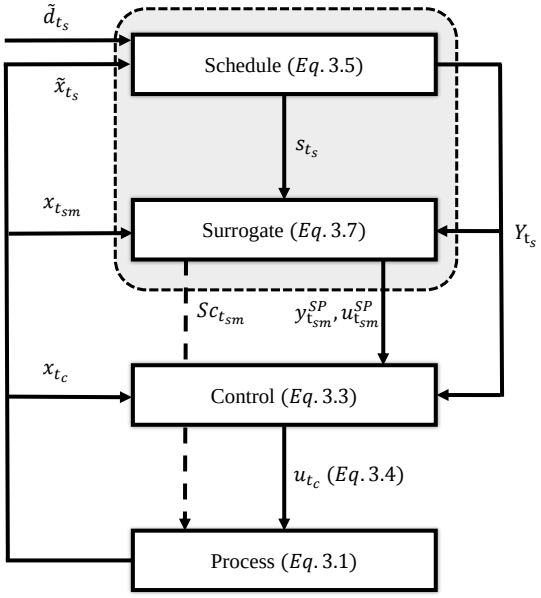


Figure 3.3: The information flow among the scheduler, surrogate model controller, and the process. The gray area indicates the overall control-aware scheduler (Reprinted with permission from [105]).

lently, one can generate 2^n mpQP problems to exhaustively enumerate all combinations of binary parameter realizations, where n is the number of binary parameters.

Step 4 – Closed-loop validation: Overall validation of the integrated schedule-control scheme is performed in a rolling horizon fashion through utilizing the maps of solutions generated with Equations 3.3, 3.5, and 3.7 simultaneously on the high fidelity model (Equation 3.1). The overall system is subjected to randomized market conditions that is updated in the time steps of the scheduler to yield the input and output trajectories in the scheduling and control levels. The interplay and the flow of information among the multiparametric scheduler, surrogate model, controller, and the process is summarized and depicted in Figure 3.3.

Note that the framework assumes an update in the disruptive events at the time steps of the schedule. Any further scheduling level disturbances in between these time steps can be addressed by reevaluating the set points through the surrogate model to remedy a potential performance degradation. The process disturbances, on the other hand, are accounted for by the controller of

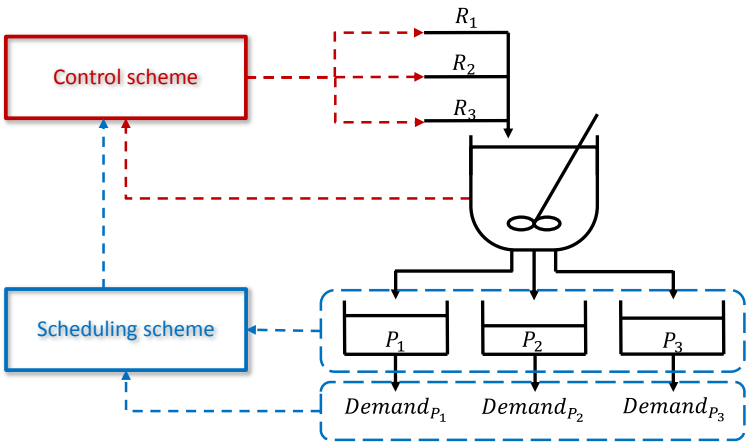


Figure 3.4: CSTR flowsheet with the online implementation of the scheduling and control schemes (Reprinted with permission from [105]).

which dynamics are embedded in the scheduler.

The following section presents the application of the framework on (i) a CSTR with three reactants and three outputs, and (ii) two CSTRs operated in parallel.

3.3 Examples

3.3.1 Single CSTR with three inputs and three outputs

65textitProblem definition: This case study, adapted from Flores-Tlacuahuac and Grossmann, (2006) [97], considers an isothermal CSTR designed to manufacture three products on a single production line, as depicted in Figure 3.4. In the figure, R_i denotes the i^{th} reactant, P_j denotes the j^{th} product, and $Demand_{P_j}$ denotes the demand rate for product P_j. The problem statement encompasses the following: foot

(i) *Given*: A high-fidelity model of the three product CSTR, unit cost for inventory, a scenario of product demands.

(ii) *Determine*: Production sequence, production rates, optimal control actions to achieve the target production rate and to reach the threshold purity.

(iii) *Objective*: Minimize the total cost comprising the inventory and transition costs.

Based on the described problem definition, the control scheme aims to determine the optimal transitions between the production periods of three products through tracking a time variant product concentration set point to maintain a certain level of purity threshold. The controller is designed to deliver this short term objective by manipulating the feed composition at the inlet of the reactor, and monitoring the states of the system. To obtain the longer term objectives, we utilize a scheduling scheme to minimize the operating and inventory costs, while satisfying a continuous demand rate for each product. The scheduler aims to determine the optimal production sequence and manufacturing time, while accounting for the inventory levels in the storage tanks and a demand scenario. The scheduling decisions are passed on to the controller as set points and operating modes. Note that different from Flores-Tlacuahuac and Grossmann, (2006) [97], this work relaxes the assumption of constant product demand rate profile, and considers a variable demand rate profile.

High-fidelity dynamic model: Three irreversible reactions take place in parallel in the CSTR reaction network given in Eq. 3.8.

$$\begin{aligned} 2R_1 &\xrightarrow{k_1} P_1 \\ R_1 + R_2 &\xrightarrow{k_2} P_2 \\ R_1 + R_3 &\xrightarrow{k_3} P_3 \end{aligned} \tag{3.8}$$

where k_1, k_2, and k_3 are the rate constants of the respective reactions. Note that production of P_1 requires only R_1, which also features as one of the raw materials of products P_2 and P_3. Hence, the given reaction network yields P_1 as a by-product during the production phases of P_2 and P_3. The by-product concentration degrades the purity of the product of interest, and needs to be accounted for by the control scheme to achieve high selectivity.

The high-fidelity model that describes the dynamic behavior of the CSTR comprises mole balances (Eq. 3.9) and power law kinetic expressions for elementary reactions (Eq. 3.10).

$$\begin{aligned} \frac{dC_{R_i}}{dt} &= \frac{Q_{R_i}C^f_{R_i} - Q_{total}C_{R_i}}{V} + \mathcal{R}_{R_i} \\ \frac{dC_{P_j}}{dt} &= \frac{Q_{total}(C_{P_j} - C_{P_j})}{V} + \mathcal{R}_{P_j} \end{aligned} \tag{3.9}$$

Table 3.2: Parameters of the high-fidelity CSTR model (Reprinted with permission from [105]).

Reaction rate constants	Value	Reactant concentration at the feed	Value
k_1	0.1	$C^f_{R_1}$	1.0
k_2	0.9	$C^f_{R_2}$	0.8
k_3	1.5	$C^f_{R_3}$	1.0

$$
\begin{aligned}
\mathcal{R}_{R_1} &= -2\mathcal{R}_{P_1} - \mathcal{R}_{P_2} - \mathcal{R}_{P_3} \\
\mathcal{R}_{R_2} &= -\mathcal{R}_{P_2} \\
\mathcal{R}_{R_3} &= -\mathcal{R}_{P_3} \\
\mathcal{R}_{P_1} &= k_1 C^2_{R_1} \\
\mathcal{R}_{P_2} &= k_2 C_{R_1} C_{R_2} \\
\mathcal{R}_{P_3} &= k_3 C_{R_1} C_{R_3}
\end{aligned}
\tag{3.10}
$$

where C denotes the concentration, Q is the volumetric flow rate, V is the volume of the CSTR, $\mathcal{R}$ is the reaction rate, superscript f denotes the feed to the CSTR, R_i and P_j are the indices for the i^{th} reactant and j^{th} product, respectively. The system parameters are given in Table 3.2.

The total volumetric flow rate is defined as the sum of reactant flow rates at the inlet of the reactor. Note that constant volume reactor is assumed, therefore the total flow rate at the inlet is equal to the total flow rate at the outlet.

$$
Q_{total} = \sum_i Q_{R_i} \tag{3.11}
$$

The inventory levels of the product of interest is as follows.

$$
\frac{dW_{P_j}}{dt} = \begin{cases} Q_{total} C_{P_j} - DR_{P_j}, \text{ if } Pur_{P_j} \geq 0.90 \\ -DR_{P_j}, \text{ if } Pur_{P_j} < 0.90 \end{cases} \tag{3.12}
$$

where W_{P_j} is the inventory level, DR_{P_j} is the demand rate, and Pur_{P_j} is the purity level in the

CSTR as defined in Equation 3.13.

$$Pur_{P_j} = \frac{C_{P_j}}{\sum_j C_{P_j}} \tag{3.13}$$

The molar fractions of the reactant flow rates are defined in Equation 3.14. Note that the molar fractions are utilized as the manipulated variables in the mpMPC control scheme, as demonstrated in the following sections.

$$\begin{aligned} a_{R_i} &= \frac{Q_{R_i}}{Q_{total}} \\ \sum_i a_{R_i} &= 1 \end{aligned} \tag{3.14}$$

Model approximation: The highly nonlinear nature of the model necessitates partitioning of the input space to capture the system dynamics with higher accuracy. Rigorous simulations of the high fidelity model suggests the partitioning of each degree of freedom available to the controller (i.e. a_{R_2} and a_{R_3}) to at least two mutually exclusive subspaces, respectively. Hence, the discrete time state space model generated in the form presented by Eq. 3.2.

Remark 3. *A significant difference of this example from the binary distillation column example presented in Chapter 2 is the multitude of operating regions in the CSTR. The binary distillation column is designed to operate at a single desired set point, as opposed to the CSTR is expected to deliver three products with a time-variant set point on product concentration based on market conditions. Therefore, we derive multiple state space models that are defined over mutually exclusive domains. The domains of the state space models with respect to the input space is given by Eq. 3.15.*

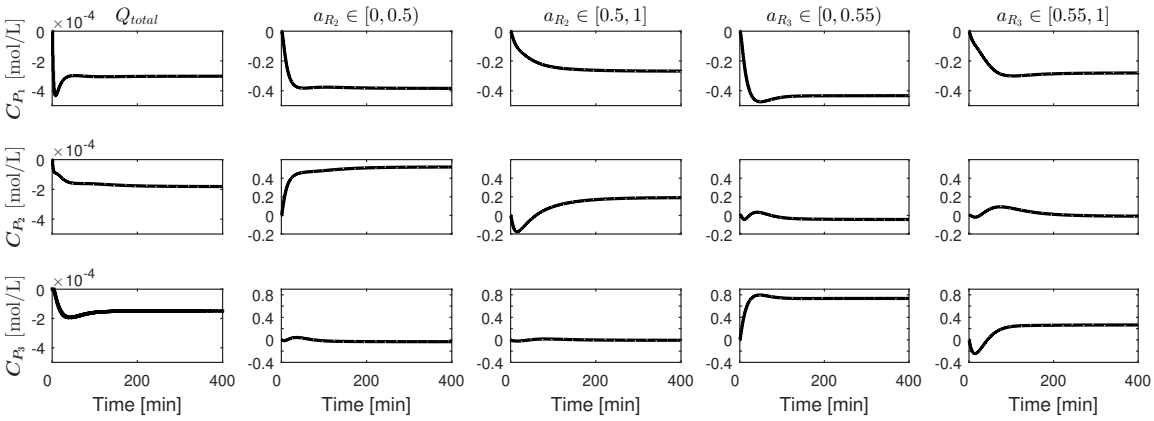


Figure 3.5: Step responses of the identified open loop model with respect to the system inputs and the scheduling variable Q_{total} (Reprinted with permission from [105]).

$$
\begin{aligned}
u_{t_c} &= \left[u_{1,t_c}, u_{2,t_c}, u_{3,t_c}, u_{4,t_c}\right]^T \\
u_{1,t_c} &= a_{R_2},\ a_{R_2} \in [0, 0.5) \\
u_{2,t_c} &= a_{R_2},\ a_{R_2} \in [0.5, 1] \\
u_{3,t_c} &= a_{R_3},\ a_{R_3} \in [0, 0.55) \\
u_{4,t_c} &= a_{R_3},\ a_{R_3} \in [0.55, 1]
\end{aligned}
\tag{3.15}
$$

where x are the identified states, u are the molar fractions of the reactant flow rates partitioned in the input space as given in Equation 3.15, d are the total volumetric flow rate (Q_{total}), and y are the product concentrations (C_{P_j}). The state space matrices are given in the Appendix A.1. Note that a_{R_1} is excluded from the manipulated variables due to the linear independence of the molar fractions.

The step and impulse responses of the open loop approximate model are stable within the range of inputs, as presented in Figures 3.5 and 3.6, respectively.

Design of the mpMPC: The formulation of the mpMPC is based on Eq. 3.3 with additional soft constraints included as presented in Eq. 3.16. The tuning of the corresponding parameters is based on heuristic MPC design methods, and the parameters are provided in Table 3.3.

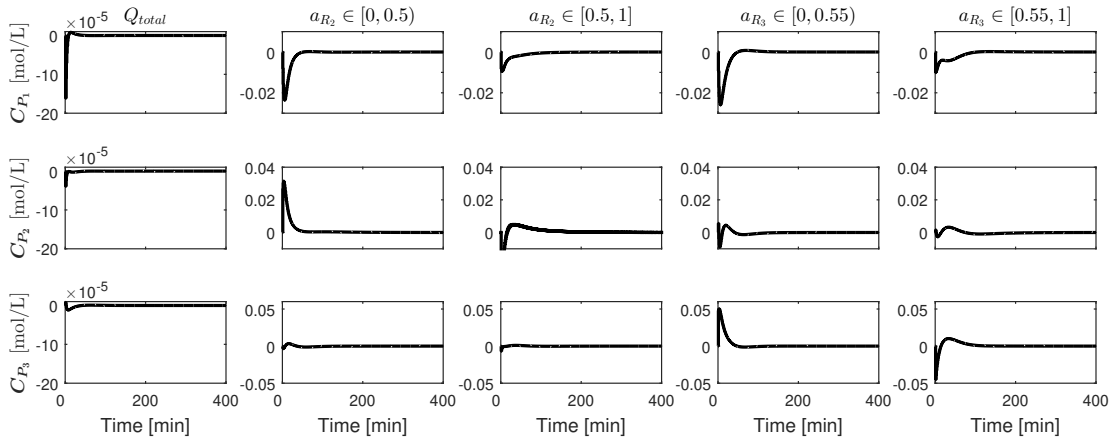


Figure 3.6: Impulse responses of the identified open loop model with respect to the system inputs and the scheduling variable Q_{total} (Reprinted with permission from [105]).

$$
\begin{aligned}
\underset{u_{t_c}, z_{t_c}, \varepsilon_{t_c}}{min} \quad & J(\theta) = \sum_{t_c=1}^{N_c} \|y_{t_c} - y_{t_c}^{SP}\|_{QR}^2 + \sum_{t_c=0}^{M_c-1} \|\Delta u_{t_c}\|_{R1}^2 + \sum_{t_c=1}^{N_c} \|\varepsilon_{t_c}\|_{P1}^2 \\
s.t. \quad & x_{t_c+1} = A_c x_{t_c} + B_c u_{t_c} + C_c S c_{t_c} \\
& \hat{y}_{t_c} = D x_{t_c} \\
& y_{t_c} = \hat{y}_{t_c} + e, \quad e = y_{t_c=0} - \hat{y}_{t_c} \\
& \underline{x} \leq x_{t_c} \leq \overline{x}, \quad \underline{y} \leq y_{t_c} \leq \overline{y}, \quad \underline{u} z_{t_c} \leq u_{t_c} \leq \overline{u} z_{t_c} \\
& \underline{u} Y_{t_c} \leq u_{t_c} \leq \overline{u} Y_{t_c}, \quad \underline{d} \leq d_{t_c} \leq \overline{d}, \quad \underline{\Delta u} \leq \Delta u_{t_c} \leq \underline{\Delta u} \\
& -y_{*,t_c} + Pur_{min} \sum_i y_{i,t_c} \leq -\varepsilon_{t_c} + \mathcal{M} Y_{t_c} \\
& 0 \leq \varepsilon_{t_c} \leq 1, z_{t_c} \in \{0, 1\} \\
& \theta = [x_{t_c=0}^T, y_{t_c=0}^T, d_{t_c=0}^T, (y_{t_c}^{SP})^T, u_{t_c=-1}^T, Y_{t_c}^T]^T \\
& \forall t_c \in \{0, 1, ..., N_c\}
\end{aligned}
\tag{3.16}
$$

where the additional terms ε is the slack variables, $P1$ is the penalty matrix, Pur_{min} is the minimum purity level required to trigger accumulation in the storage tanks, y_* is the concentration of the product of interest, and $\mathcal{M}$ is a big-M parameter. The binary switch parameter Y determined by an upper level decision maker dictates the product of interest, and binary switch variable z de-

Table 3.3: Tuning parameters for the mpMPC of the single CSTR example (Reprinted with permission from [105]).

mpMPC design parameters	Value
N_c	6
M_c	2
QR	$\begin{bmatrix} 10^2 & 0 & 0 \\ 0 & 10 & 0 \\ 0 & 0 & 10 \end{bmatrix}$
$R1$	50
$P1$	90
$\underline{y}$	$[0,0,0]^T$
$\overline{y}$	$[1,1,1]^T$
$\underline{u}$	$[0,0,0]^T$
$\overline{u}$	$[1,1,1]^T$
$\underline{d}$	0
$\overline{d}$	500

termines the optimal input subspace. The soft constraints are constructed on y_* via slack variables ε to minimize the transition time by penalizing any production below the threshold purity level throughout the output horizon. Therefore, the non-negative slack variables ε contribute to the objective function if and only if the purity of the product of interest is below the threshold. Note that any process disturbances, such as reactant concentrations at the feed stream, can be easily incorporated in the control scheme without modifying the overall framework by simply introducing them as additional parameters.

The optimization problem given in Equation 3.16 is reformulated as a mpMIQP problem and solved via the POP toolbox to generate the map of optimal control actions as affine functions of the system parameters. The explicit expression of the control action is designed to (i) track a set point determined by an upper level decision maker, (ii) adapt proactively to changing operating modes (i.e. shifting between different products), and (iii) minimize the transition time by penalizing impure production periods.

Note that the mpMPC formulation utilizes a single state space model with piecewise affine inputs that are selected via binary switch variables, z_{t_c}. Therefore, the control scheme single-

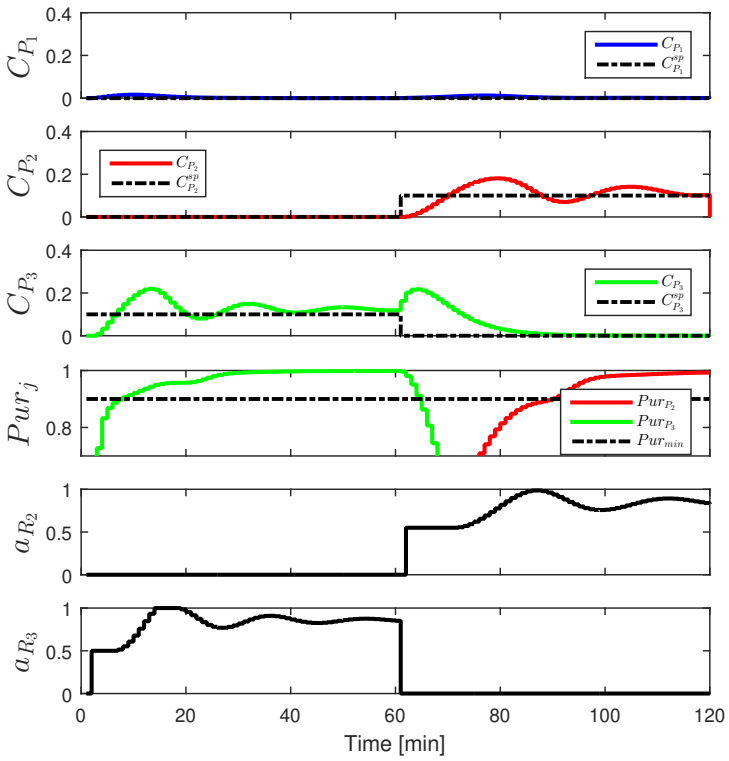


Figure 3.7: Closed-loop validation of the mpMPC against the high-fidelity model for the single CSTR example (Reprinted with permission from [105]).

handedly recognizes the dynamics of the transitions between the production periods. Although the stability of the system under such transitions is left outside the scope of this study, it can be further investigated following the approach proposed by Grieder *et al.* (2004) [196].

Closed-loop validation: The control scheme is validated by exhaustive testing against the high-fidelity dynamic model under various scheduling decisions. Figure 3.7 presents a 2 h closed-loop operation with two distinct operational modes. The process starts from zero product concentration and goes through a shift from the production of P_3 to production of P_2 at $t = 60\ min$. This shift is manually enforced by changing the concentration set points from $y^{SP} = [0, 0, 0.1]^T$ to $y^{SP} = [0, 0.1, 0]^T$.

The closed-loop simulation in Figure 3.7 validates the mpMPC as it (i) tracks the set points of three product concentrations, (ii) handles operations at different production modes, (iii) prioritizes

purity satisfaction to minimize transition time, and (iv) maintains feasible operation by keeping the system within specified bounds. Note that the entirety of the closed loop simulations uses only one mpMPC scheme in both the production and the transition periods. Therefore, the controller parameters are tuned regarding every possible transition between the products.

High-fidelity model with the mpMPC embedded: The initial high-fidelity model given in Eq. 3.9-3.14 is integrated with the derived control scheme in the form of Eq. 3.4.

Model approximation: To keep the example tangible, only the mole balance around the storage tanks is considered in the upper level schedule, while the dynamics of the CSTR is accounted for in the lower level surrogate model formulation. The bilinear $Q_{total}C_{P_j}$ term in Eq. 3.12 results into a non-convex mpMINLP problem, for which only approximate solution algorithms exist. Hence, we postulate a mpMILP problem, for which POP toolbox features an exact algorithm, via replacing Eq. 3.12 with Eq. 3.17.

$$\frac{dW_{P_j}}{dt} = F_{P_j} - DR_{P_j} \tag{3.17}$$

where F_{P_j} is the molar product flow rate at the exit of the CSTR. Having merely linear terms in Eq. 3.17 enables the formulation of a mpMILP in the subsequent step.

The lower level surrogate model, on the other hand, is identified via MATLAB® System Identification Toolbox™ as described in the previous section. Three surrogate models are derived for three distinct operational modes (provided in Appendix A.1 with their respective step and impulse responses).

Design of the scheduler and the surrogate model: The scheduler for this problem is designed to minimize the inventory cost, while satisfying continuous demand rates for the three products forecasted through the scheduling horizon, as presented in Eq. 3.18.

Table 3.4: System parameters for the scheduler of the single CSTR example (Reprinted with permission from [105]).

System parameters	Value
N_s	3
α $[\$/h.mol]$	$[1.0, 1.5, 1.8]^T$
Δt_s $[min]$	60
$\underline{F}$	$[0, 0, 0]^T$
$\overline{F}$	$[50, 50, 50]^T$
$\underline{W}$	$[0, 0, 0]^T$
$\overline{W}$	$[50, 50, 50]^T$
$\underline{D}$	$[0, 0, 0]^T$
$\overline{D}$	$[60, 60, 60]^T$

$$
\begin{aligned}
\underset{F_{j,t_s}, Y_{P_j,t_s}}{min} \quad & J(\theta) = \sum_{j=1} \sum_{t_s=1}^{N_s} \alpha_{P_j}^T W_{t_s} \\
s.t. \quad & W_{P_j,t_s+1} = W_{P_j,t_s} + \Delta t_s F_{j,t_s} - \Delta t_s DR_{P_j,t_s} \\
& \sum_{j=1} F_{j,t_s} = F_{total,t_s} \\
& \sum_{j=1} Y_{P_j,t_s} = 1 \\
& \underline{F} Y_{P_j,t_s} \leq F_{j,t_s} \leq \overline{F} Y_{P_j,t_s} \\
& \underline{W} \leq W_{P_j,t_s} \leq \overline{W} \\
& \underline{DR} \leq DR_{P_j,t_s} \leq \overline{DR} \\
& \theta = [W_{P_j,t_s=0}^T, DR_{P_j,t_s}^T]^T \\
& Y_{P_j,t_s} \in \{0, 1\}, \ \forall t_s \in \{0, 1, ..., N_s\}
\end{aligned}
\tag{3.18}
$$

where Y_{P_j,t_s} denotes the selected product P_j to be manufactured at time t_s, F_{j,t_s} is the molar product flow rate, Δt_s is the sampling time for the schedule. Note that Equation 3.17 is discretized into time steps Δt_s. The system parameters for Equation 3.18 are given in Table 3.4.

The bridge between the mpMPC and the scheduler derived in Equations 3.16 and 3.18 is constructed based on Equation 3.7. Analogous to the mpMPC, the surrogate model also features the

soft constraints to enforce a threshold purity level.

$$
\begin{aligned}
\underset{Q_{total,t_{sm}},C^{SP}_{P_j,t_{sm}},\varepsilon'_{t_{sm}}}{min} \quad & J(\theta) = \sum_{t_{sm}=0}^{M_{sm}} \|Q_{total,t_{sm}} - \tilde{Q}_{total,t_{sm}}\|^2_{R'} + \sum_{t_{sm}=1}^{N_{sm}} \|\varepsilon'_{t_{sm}}\|^2_{P1'} \\
s.t. \quad & Equations\ A.1 - A.3 \\
& \tilde{Q}_{total,t_{sm}} = \frac{F_{total,t_{sm}}}{C_{P_*,t_{sm}=0}} \\
& \underline{y} \leq y_{t_{sm}} \leq y_{max} \\
& \underline{Q}_{min} \leq Q_{total,t_{sm}} \leq Q_{max} \\
& \underline{C^{SP}} \leq C^{SP}_{P_j,t_{sm}} \leq C^{SP}_{max} \\
& -y_{*,t_{sm}} + Pur_{min} \sum_i y_{i,t_{sm}} \leq -\varepsilon'_{t_{sm}} \\
& 0 \leq \varepsilon'_{t_{sm}} \leq 1 \\
& \theta = \left[x^T_{t_{sm}}, \frac{F_{total,t_{sm}}}{C_{P_*,t_{sm}=0}} \right]^T \\
& \forall t_{sm} \in \{0, 1, ..., N_{sm}\}
\end{aligned}
\tag{3.19}
$$

Note that the formulation given in Equation 3.19 is only valid for the product of interest. Hence, three separate formulations are constructed for each product. Tuning of the surrogate model parameters is based on heuristic decisions that yield a desirable performance in the closed loop validation, and the parameters are given in Table 3.5.

Closed-loop validation of the overall scheme: Closing the loop of the CSTR is performed via testing the scheduling and control scheme against the high fidelity model. Figure 3.8 presents a 12 h operation with the scheduler, the surrogate model, and the controller operating in tandem with the dynamic model while no specific knowledge of the demand profile assumed. A sample of the explicit simultaneous decisions is demonstrated in Table 3.6, where the functional form of the scheduling and control actions at $t = 60\ min$ is shown. The scheduler (i) maintains low inventory levels, and (ii) adapts to the changes in the demand profile, while satisfying the continuous demand

Table 3.5: System parameters for the surrogate model of the single CSTR example (Reprinted with permission from [105]).

System parameters	Model 1	Model 2	Model 3
N_{sm}	10	10	10
M_{sm}	1	1	1
$\Delta t_{sm}\ [min]$	6	6	6
R'	10^3	$\begin{bmatrix} 10^{-4} & 0 \\ 0 & 10^{-1} \end{bmatrix}$	$\begin{bmatrix} 10^{-4} & 0 \\ 0 & 10^{-1} \end{bmatrix}$
$P1'$	10^4	10^6	10^8
$y_{min}\ [mol/L]$	$[0,0,0]^T$	$[0,0,0]^T$	$[0,0,0]^T$
$y_{max}\ [mol/L]$	$[1,1,1]^T$	$[1,1,1]^T$	$[1,1,1]^T$
$Q_{min}\ [L/min]$	0	0	0
$Q_{max}\ [L/min]$	500	500	500
$C^{SP}_{min}\ [mol/L]$	$[0,0,0]^T$	$[0,0,0]^T$	$[0,0,0]^T$
$C^{SP}_{max}\ [mol/L]$	$[1,1,1]^T$	$[1,1,1]^T$	$[1,1,1]^T$

rate. Due to the rolling horizon strategy, the schedule is updated at every discretization step Δt_s, with the current inventory level and the new demand profile N_s time steps into the future. Note that the resultant production sequence is different from a cyclic schedule reported in Flores-Tlacuahuac and Grossmann [97] and Zhuge and Ierapetritou [101], since the demand rates in this work are time variant.

Figure 3.9 presents a snapshot of the first 6 h of operation, focusing on the lower level surrogate model decisions. The volumetric feed flow rate from the schedule and the surrogate model are juxtaposed in Figure 3.9a to emphasize the corrective actions of the latter. During the transition between production regimes, the surrogate model saturates Q_{total} at its upper bound to purge the previous product left in the reactor. The transitions can also be monitored from the product purities presented in Figure 3.9c. The surrogate model and the mpMPC operate in tandem to drive the system above the threshold purity level. The transitions to product P_2 specifically show that the integrated schedule and control scheme prioritizes the purity satisfaction to minimize the transition time.

Note the following:

- The explicit expressions for the optimal scheduling decisions enable rescheduling with a

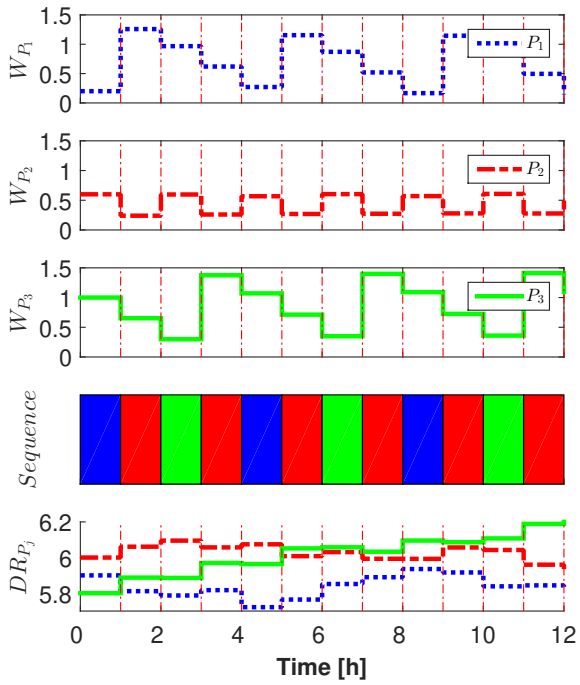


Figure 3.8: Closed-loop validation of the scheduling scheme for the single CSTR example (Reprinted with permission from [105]).

Table 3.6: Optimal scheduling and control decisions at $t = 60\ min$ (Reprinted with permission from [105]).

Decision variable	Affine expression
$F_{total,t_s=0}$	$= -16.7W_2 + DR_{t_s=0,2} + DR_{t_s=1,2}$
$F_{total,t_s=1}$	$= -16.7W_3 + DR_{t_s=0,3} + DR_{t_s=1,3} + DR_{t_s=2,3}$
$F_{total,t_s=2}$	$= DR_{t_s=2,2}$
Q_{total}	$= 500$
$C_{P_1}^{SP}$	$= 0$
$C_{P_2}^{SP}$	$= 0.91C_{P_1} - 0.01C_{P_2} + 0.02$
$C_{P_3}^{SP}$	$= 0$
a_{R_1}	$= 0$
a_{R_2}	$= 0.55$
a_{R_3}	$= 0$

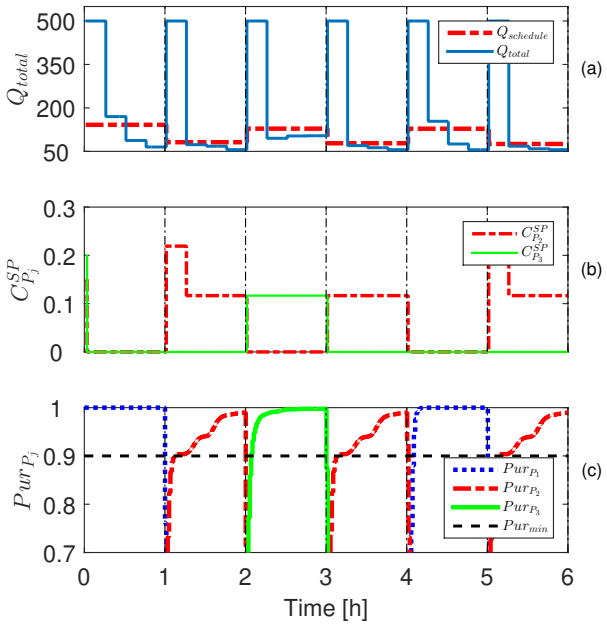


Figure 3.9: Closed-loop simulation of the CSTR for the first 6 h of operation for the CSTR example. (a) volumetric flow rate determined by the scheduler, and the corrected action of the surrogate model, (b) product concentration set points, (c) product purities (Reprinted with permission from [105]).

small computational cost when disruptive events occur in the product demands.

- The transition time is not determined explicitly by the integrated scheduling and control scheme, but is minimized through soft constraints in the surrogate model and controller formulations. The non-negative slack variables ε_{t_c} and $\varepsilon'_{t_{sm}}$ in Eq. 3.16 and 3.19 are nonzero only if the product concentration of interest is below the threshold level, and contribute to the objective function $J(\theta)$ proportional to $P1$ and $P1'$, respectively. A more accurate approach would be allocating every time step for all products with binary variables to determine whether the purity threshold is satisfied. However, employing such a large number of binary variables in a multiparametric programming problem results in an exponential increase in the computational burden. Hence, we alleviate this problem via the soft constraint formulation.

- The heavy penalty terms for purity satisfaction in the surrogate models result in steep changes

in Q_{total} during the transitions, as observed in Figure 3.9. The upper level schedule is unable to make such corrective decisions due to its large time step. The lower level surrogate model provides time varying targets and set points for the controller due to the embedded closed loop dynamics in its formulation.

- Due to the strong nonlinearity of the high-fidelity model, the input space is partitioned as presented in Equation 3.15. Finer partitions will yield more accurate controllers at the expense of increased computation time to generate the offline maps of optimal actions.

- Utilizing the maps of optimal solutions for the control, surrogate model, and schedule actions eases the online implementation. Calculation of the optimal actions is reduced from an online optimization problem to a simple look-up table algorithm and evaluation of an affine function.

3.3.2 Two CSTRs operating in parallel

Problem definition: This case study extends the CSTR example from Section 3.3.1 to encompass two identical CSTRs operating in parallel. Due to the identical design of the reactors, the mpMPC and the surrogate model driving the closed-loop system are identical as well. Hence, the derivation of their formulations and attaining the explicit maps of solutions are omitted.

Cooperative operation of independent reactors requires a centralized scheduling scheme to allocate the production tasks on different reactors. However, the identical nature of the two closed-loop system dynamics creates a multiplicity of solutions, as the reactors are indistinguishable to the upper level schedule. Hence, the scheduling formulation presented in Eq. 3.18 is modified to (i) account for the previous production regime as an additional uncertain parameter, and (ii) penalize transitions between consecutive production regimes to break multiple solutions. Inclusion of the retrospective information provides a distinction between the reactors, eliminating any redundant transitions between the products. The mathematical representation of the described modification is provided in Eq. 3.20.

$$\Gamma_s = \sum_{t_s=0}^{N_s} \psi \left| Y_{P_j,t_s} - Y_{P_j,t_s-1} \right| \tag{3.20}$$

where ψ is a very small number that virtually penalizes the changes in the operational mode. Eq. 3.20 can be reformulated as follows to maintain the linear structure of the scheduling problem.

$$\begin{aligned} & \Gamma_s = \sum_{t_s=0}^{N_s} \psi^T \bar{Y}_{t_s} \\ s.t. \quad & Y_{P_j,t_s} - Y_{P_j,t_s-1} \leq \bar{Y}_{t_s} \\ & -Y_{P_j,t_s} + Y_{P_j,t_s-1} \leq \bar{Y}_{t_s} \\ & 0 \leq \bar{Y}_{t_s} \leq 1 \end{aligned} \tag{3.21}$$

where $\bar{Y}_{t_s}$ is an auxiliary variable.

Design of the scheduler: The scheduling formulation presented in Eq. 3.18 is extended to account for multiple production lines, and modified with Eq. 3.21 to eliminate multiple solutions, which gives Eq. 3.22.

$$
\begin{aligned}
\underset{F_{j,t_s,l},Y_{P_j,t_s,l},\bar{Y}_{t_s,l}}{min} \quad & J(\theta) = \sum_{j=1}^{} \sum_{t_s=1}^{N_s} \alpha_{P_j}^T W_{P_j,t_s} + \sum_{l=1}^{N_{CSTR}} \sum_{t_s=0}^{N_s} \psi^T \bar{Y}_{t_s,l} \\
s.t. \quad & W_{P_j,t_s+1} = W_{P_j,t_s} + \sum_{l=1}^{N_{CSTR}} \Delta t_s F_{j,t_s,l} - \Delta t_s DR_{P_j,t_s} \\
& \sum_{j=1} F_{j,t_s,l} = F_{total,t_s,l} \\
& \sum_{j=1} Y_{P_j,t_s,l} = 1 \\
& Y_{P_j,t_s,l} - Y_{P_j,t_s-1,l} \leq \bar{Y}_{t_s,l} \\
& - Y_{P_j,t_s,l} + Y_{P_j,t_s-1,l} \leq \bar{Y}_{t_s,l} \\
& 0 \leq \bar{Y}_{t_s,l} \leq 1 \\
& \underline{F} Y_{P_j,t_s,l} \leq F_{j,t_s,l} \leq \overline{F} Y_{P_j,t_s,l} \\
& \underline{W} \leq W_{P_j,t_s} \leq \overline{W} \\
& \underline{DR} \leq DR_{P_j,t_s} \leq \overline{DR} \\
& \theta = [W_{P_j,t_s=0}, DR_{P_j,t_s}, Y_{P_j,t_s=-1,l}]^T \\
& Y_{P_j,t_s,l} \in \{0,1\},\ \forall t_s \in \{0,1,...,N_s\}, \forall l \in \{1,2,...,N_{CSTR}\}
\end{aligned}
\tag{3.22}
$$

where the additional weight ψ is tuned to be 0.001, and the number of the CSTRs, N_{CSTR}, is 2 by the problem definition.

Closed-loop validation: The validation of the overall scheduling and control scheme is presented in Figure 3.10. The scheduler, the surrogate model, and the controller are operated in tandem with the high fidelity model for 12 h under a randomized demand profile. The integrated scheduling and control scheme delivers the additional task to coordinate multiple reactors to operate in parallel while satisfying the continuous demand rate. The inclusion of Eq. 3.21 in the objective function breaks the symmetry between the reactors and coordinates the production sequence. Consequently, uninterrupted manufacturing of the product of interest is maintained without redundant shifts between the reactors.

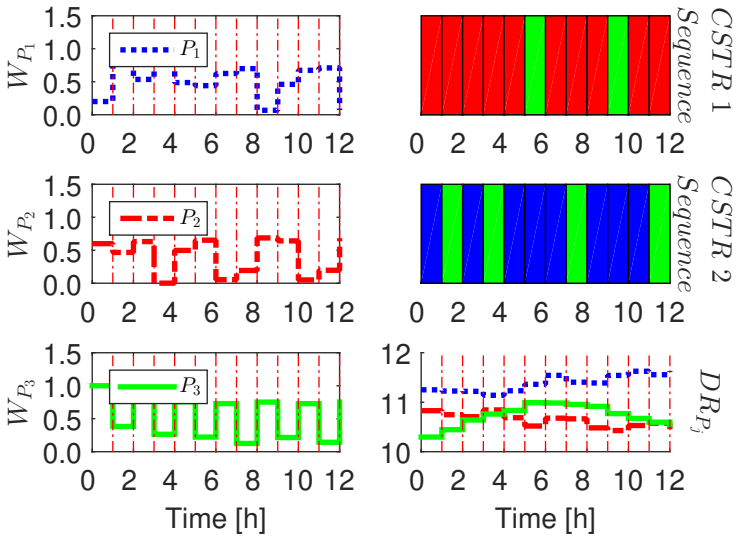


Figure 3.10: Closed-loop validation of the integrated scheduling and control scheme on two CSTR example (Reprinted with permission from [105]).

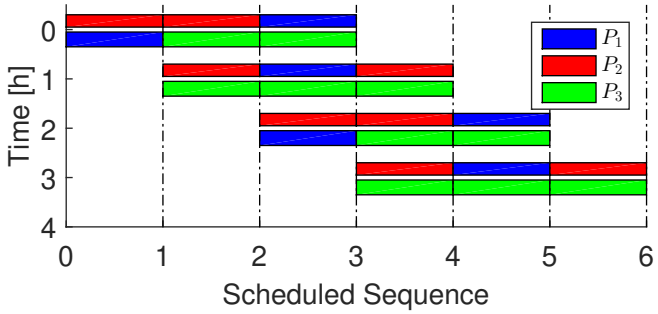


Figure 3.11: Realization of the schedule with time for the two CSTR example. Top bars and bottom bars represent CSTR 1 and CSTR 2, respectively (Reprinted with permission from [105]).

Figure 3.11 presents the evolution of the schedule with time for the first 4 h of operation. Note that the demand scenario is updated every hour in a rolling horizon manner, allowing rescheduling of the production sequence and the target quantities by utilizing the offline maps of optimal scheduling actions.

3.4 Conclusions

In this chapter, a systematic framework was presented to integrate process scheduling and control in continuous systems via multiparametric programming. Optimal scheduling and control actions were derived simultaneously based on a single high fidelity model. We take advantage of the synergistic interactions between the two decision making mechanisms to yield offline maps of optimal operations as explicit affine expressions at both long and short terms of a process. The generic structure of the framework renders it suitable for a software prototype towards enterprise-wide optimization.

This chapter aims to increase the operability, flexibility, and profitability of process systems through improving the scheduling and control decisions. Nevertheless, the processes with comparable capital and operating costs necessitates the consideration of the design aspect simultaneously with the scheduling and control. Hence, the next chapter focuses on the unification of the design, scheduling, and control actions, simultaneously.

4. INTEGRATION OF PROCESS DESIGN, SCHEDULING, AND CONTROL[1]

4.1 Introduction

Process design decisions, such as equipment selection and sizing, span the widest time-scale in the functional hierarchy of a chemical process, and they are typically established by solving a steady-state design optimization problem [79]. Operational decisions such as scheduling and control are usually assumed to take a nominal value to make the problem complexity tractable [170]. However, rapidly changing market conditions and process disturbances often force the system to operate under a wide range of operating conditions, which may render the steady-state process design dynamically infeasible. Design optimization under such operational uncertainties have been extensively investigated in the literature by considering feasibility, flexibility, stability, controllability, and resilience metrics [45, 148, 88]. However, direct inclusion of the operational dynamics in the design optimization is rather limited. Terrazas-Moreno et al. (2008) [4] and Patil et al. (2015) [5] presented MIDO based frameworks to account for the economical decisions in open loop processes. Koller et al. (2018) [7] developed a stochastic back-off algorithm for the simultaneous consideration of the design and operational optimization problems in the context of closed loop operations, where the algorithms was showcased on a multiproduct CSTR.

In Chapters 2 and 3, a multiparametric programming based approach was introduced and applied to integrate (i) the design and control, and (ii) scheduling and control problems. In this chapter, the previously introduced approaches are combined to yield a unified theory and framework to integrate the design, scheduling, and control problems. We explicitly map the upper level layer decisions on the lower levels by multiparametric programming. The explicit expressions at the lower level layers enable their representation in the upper level problems. In other words, the control problem is derived as a function of design and scheduling decisions, and similarly the scheduling decisions are design dependent, and aware of the controller dynamics. These explicit

[1]Portions of this chapter have been adapted from Burnak, B., Diangelakis, N.A., Katz, J., Pistikopoulos, E.N., Integrated process design, scheduling, and control using multiparametric programming, Computers & Chemical Engineering 2019, 125, pp. 164-184, *Special Issue* with permission from Elsevier.

scheduling and control maps allow for an exact implementation in a design optimization problem. Furthermore, we introduce a design dependent surrogate model formulation to bridge the time scale gap between the schedule and the control problems, which is also solved offline.

The remainder of the chapter is organized as follows. Section 4.2 defines the integration problem that is addressed in this study, and describes the proposed framework to approach the problem. The framework is showcased in Section 4.3 on systems of reactors introduced in Section 3.3 and residential combined heat and power (CHP) units. Lastly, Section 4.4 presents concluding remarks and future directions.

4.2 Integration of design, scheduling, and control via multiparametric optimization

Problem definition: We consider a generic process where the interactions between the long term (design), middle term (schedule), and short term (control) decisions are sufficiently significant to impact the feasibility and the optimality of each individual decision. Therefore, we define the following problem that encapsulates all three decisions simultaneously.

(i) *Given*: A high fidelity model based on first principles or data-driven modeling techniques that accurately captures the dynamics of the system, any physical limitations of the system due to process safety considerations or product specifications, unit costs for design, raw material, energy, and inventory, revenue for unit product, and an accurate demand forecast.

(ii) *Determine*: Production sequence throughout an operating horizon, closed loop control strategy that delivers the product specifications, set points for the operation tailored for the dynamics of the closed loop strategy, size of the processing equipment that ensures operability of the process.

(iii) *Objective*: Minimize the operating and capital costs.

Therefore, the defined problem can be formulated as the initial integrated MIDO problem, introduced in Eq. 1.11. Solving this problem comprises the challenges in the integrated design and control, and scheduling and control problems, which have been discussed and addressed in

Chapters 2 and 3. The proposed solution strategy to the overall integrated problem hence utilizes the key pieces of the previously introduced frameworks, described as follows.

(i) Develop an offline control policy that takes into account the different process dynamic stemming from the selection of the unit design (see Chapter 2) and online economical decisions (see Chapter 3).

(ii) Derive a scheduling policy based on the closed loop behaviour of the system and the design variables (see Chapter 3).

(iii) Determined the design that minimizes the capital and operating costs for a given time period by utilizing the offline control and scheduling policies simultaneously (see Chapter 2).

Given that the foundation of the framework has been outlined in the previous chapters, we will showcase the framework on (i) the CSTR examples introduced in Section 3.3.1 and 3.3.2, and (ii) a small system of residential combined heat and power units. The interested reader can appeal to Appendix A.2 for a detailed walk-through of the framework.

4.3 Examples

4.3.1 Single CSTR with three inputs and three outputs

In this section, the CSTR example introduced and defined in Section 3.3.1 is revisited to account for the design variables along with the operational optimization problem. Therefore, the problem definition of the example is defined as follows.

(i) *Given*: A high-fidelity model of the three product CSTR, unit inventory costs, a functional expression for the CSTR fixed cost, a scenario of product demands.

(ii) *Determine*: Volume of the CSTR, production sequence, production rates, optimal reactant volumetric flow rates to achieve the target production rate and to reach the threshold purity.

(iii) *Objective*: Minimize the sum of operating and capital costs.

The objective in the problem definition can be achieved by determining the reactor design, production schedule, and closed loop dynamics that minimize the wasted raw materials and processing time. Therefore, (i) the controller is expected to deliver optimal transitions between all operating points determined by the scheduler, (ii) the scheduling decisions have to minimize the operating costs while accounting for the closed loop dynamics, and (iii) the reactor must be large enough to remain feasible throughout the entire operation, while avoiding overdesign to minimize the capital costs.

We use the same high fideltiy model to simulate the dynamics of the CSTR that was introduced in Section 3.3.1. However in this example, the dynamics introduced by the design variable (reactor volume) needs to be included in the approximate model. Therefore, the reduced order state space model is developed as given by Eq. 4.1.

$$\begin{aligned} x_{t_c+1} &= Ax_{t_c} + B\begin{bmatrix} u_{1,t_c} \\ u_{2,t_c} \\ u_{3,t_c} \\ u_{4,t_c} \end{bmatrix} + C\begin{bmatrix} Q_{total,t_c} \\ V \end{bmatrix} \\ \hat{C}_{i,t_c} &= Dx_{t_c}, \quad i \in P \end{aligned} \tag{4.1}$$

where x are the identified states, u_{k,t_c} is the k^{th} partitioning of input u at time instance t_c, Q_{total} is the volumetric flowrate at the feed, V is the volume of the reactor, and $\hat{C}_i$ is the i^{th} product.

The developed approximate model is incorporated into the mpMPC problem described in Eq. 3.3, with the mere difference of the design dependence of the approximate model. Note that, (i) the input space is partitioned to mutually exclusive domains to capture a wider range of operating regions, and (ii) soft constraints are included to minimize the transition time at the control level.

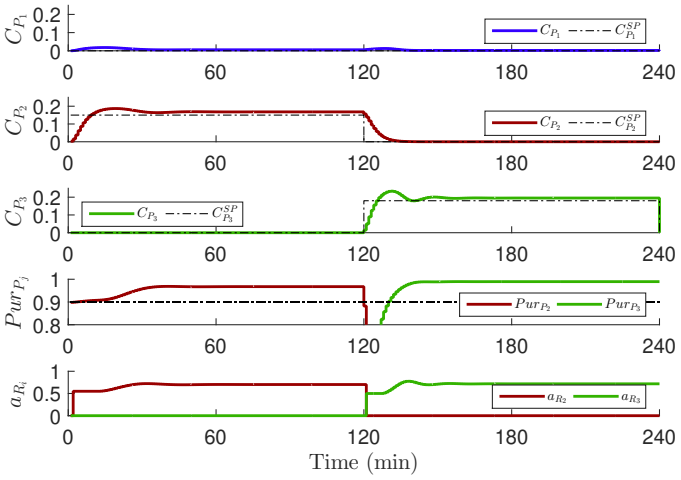
The developed mpMPC is validated against the high-fidelity model, under a range of scheduling decisions and design options. Figure 4.1 presents 4 h closed loop simulations for two reactor volumes ($V_1 = 0.4m^3$, $V_2 = 1.0m^3$). The process undergoes a step change from P_2 to P_3 after 2 h of operation to test the validity of the control scheme under different scheduling decisions and

design configurations. Note that all operations are governed by a single explicit control law that is a function of the design and scheduling decisions.

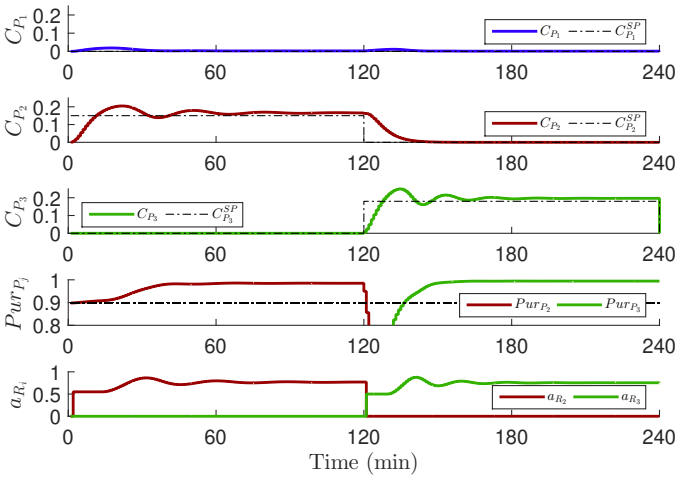
The closed loop simulations presented in Figure 4.1 shows that the developed control scheme is suitable for a range of scheduling and design options. The control scheme (i) achieves effective set point tracking for all three products simultaneously, (ii) minimizes transition time by prioritizing the purity satisfaction, (iii) recognizes the dynamics introduced by different scheduling decisions and design configurations, and (iv) maintains the operation within the inherent/imposed bounds of the system.

The validated explicit control law is integrated into the original high fidelity model, which is used to simulate the closed loop process. Based on the dynamics of the new process dynamics, we develop approximate models that include the design variable as a measured disturbance. The developed approximate models for the closed loop process are provided in Appendix A.1 along with their step responses. We utilize the upper level schedule introduced in Eq. 3.5 for the long term economical decisions.

After deriving all the design dependent operational strategies, the controller, surrogate model, and the scheduler are operated simultaneously on the high fidelity model under a range of design options and product demand variations. Figure 4.2 showcases the closed loop profiles for 12 hours at the lower bound ($V_1 = 0.4m^3$) and the upper bound ($V_2 = 1.0m^3$) of the design range. Note that the same design dependent offline strategies are used in two reactors. The demand profiles for the products are randomly regenerated every hour, and the scheduling decisions are updated in a rolling horizon manner. The closed-loop simulations validate that the integrated scheduling and control scheme (i) maintains low inventory levels in the storage tanks, (ii) reactively adapts to changes in the demand profile, (iii) is applicable for a range of different design options. A sample of the offline scheduling and control decisions is demonstrated in Table 4.1, where a snapshot of the online operation of the large CSTR at $t = 5h$ is tabulated. Such explicit expressions are available for the range of design decisions, and will be used for design optimization described as follows.

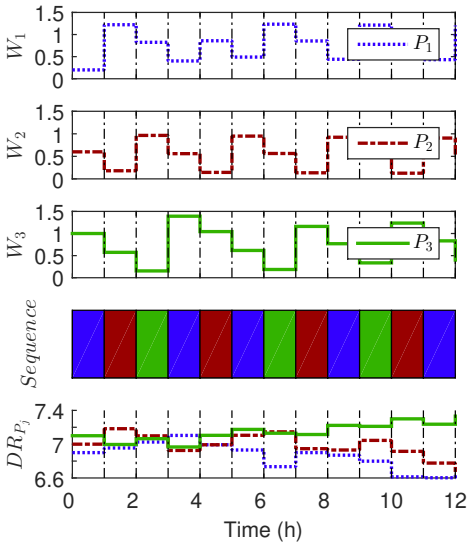


(a) $V_1 = 0.4m^3$

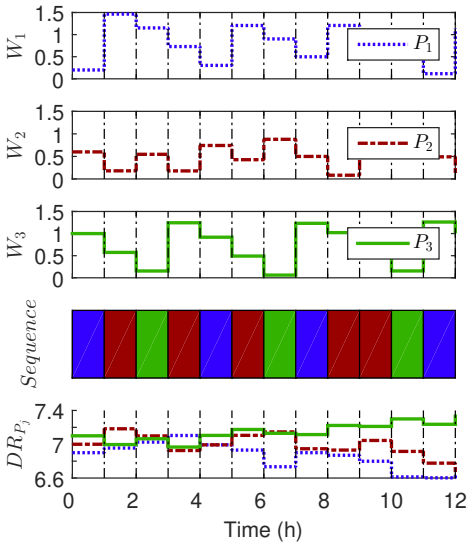


(b) $V_2 = 1.0m^3$

Figure 4.1: Single CSTR example - Step change in set points in two reactors with different volumes (Reprinted with permission from [8]).



(a) $V_1 = 0.4m^3$



(b) $V_2 = 1.0m^3$

Figure 4.2: Single CSTR example – Closed-loop validation of the integrated scheduling and control scheme in two reactors with different volumes (Reprinted with permission from [8]).

Table 4.1: Single CSTR example – An illustration of the offline map of receding horizon policies at $t = 5h$ for the large CSTR ($V_2 = 1.0m^3$). Observe that the volume of the reactor has a direct impact on the control action for this particular instance (Reprinted with permission from [8]).

Decision variable	Affine expression
$F_{3,t=0}$	$= -16.7W_3 + DR_{3,t=0} + DR_{3,t=1h} + DR_{3,t=2h}$
$F_{2,t=1h}$	$= -16.7W_2 + DR_{2,t=0} + DR_{2,t=1h} + DR_{2,t=2h}$
$F_{1,t=2h}$	$= -16.7W_1 + DR_{1,t=0} + DR_{1,t=1h} + DR_{1,t=2h}$
$Q_{total,t=0}$	$= 500$
$CP^{SP}_{2,t=0}$	$= 0.91(CP_{1,t=0} - 0.003) - 0.007(CP_{2,t=0} - 0.14) - 0.12$
a_1	$= 0.45 - 6 \times 10^{-3}V$
a_2	$= 0.55 + 6 \times 10^{-3}V$

The validated offline scheduling and control strategies are embedded in the overall MIDO problem given in Eq. 1.11 in the gPROMS environment. The capital investmentment for the reactor is given by Eq. 4.2 [197].

$$C_e = a + bV^n \tag{4.2}$$

where C_e is the annualized reactor cost, and a, b, n are cost parameters given in Appendix A, along with the cost escalation indexes for year 2018. The minimum total annual cost is found as $\$330k/yr$ at $V = 0.69m^3$. Note that the scheduling and control strategies yield feasible operation for the optimal reactor volume as a result of their design dependence. Therefore, treating the design, scheduling, and control problems simultaneously ensures the operability of the system, as the MIDO problem comprises the exact closed loop strategies that will be used online during the operation.

4.3.2 Two CSTRs operating in parallel

This case study presents an extension of the single CSTR example discussed in Section 4.3.1 to two CSTRs operating in parallel. The exact same control strategy and the surrogate model formulations are employed because the open loop dynamics of the system remains unchanged. The cooperative operation of the two CSTRs is maintained by a centralized scheduler that allocates the

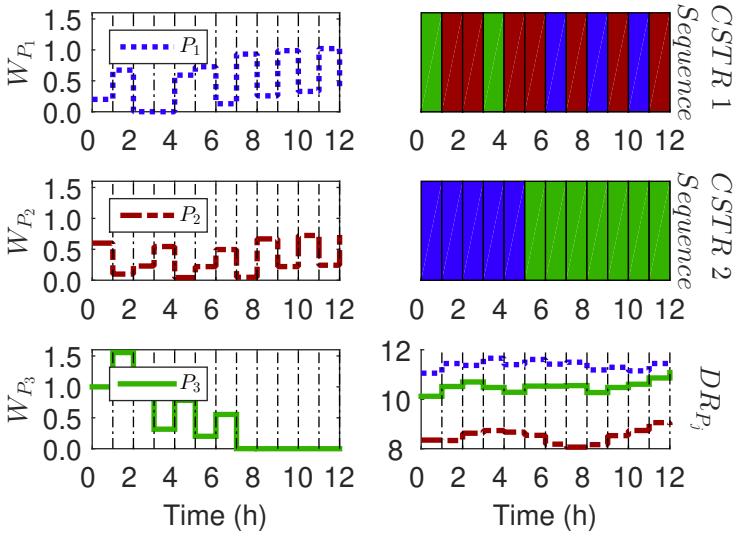


Figure 4.3: Two CSTRs in parallel - Closed loop validation of the generalized scheduling scheme in two reactors operating in parallel. The volumes of the reactors are $V_1 = 0.4m^3$ and $V_2 = 1.0m^3$, respectively (Reprinted with permission from [8]).

production tasks on the reactors based on their volumes and their production regimes at a given time. This scheduler is based on the formulation presented by Eq. 3.5, and is omitted in this section.

The generalized offline scheduling scheme is validated against the high fidelity model of the two reactor system. Figure 4.3 showcases a scenario with one small reactor ($V_1 = 0.4m^3$) and one larger reactor ($V_2 = 1.0m^3$) operated in parallel. The integrated scheduling and control scheme is able to drive the inventory level of the most costly product, W_{P_3}, close to zero by assigning it to the larger reactor. The large reactor is capable of satisfying the demand on P_3 standalone, and the small reactor has a faster transition rate because of the lower retention time. Therefore recognizing the closed loop dynamics and the capacity of the reactors, the integrated schedule assigns the costly product, P_3, to the large reactor, and alternates the production between P_1 and P_2 in the small reactor.

Finally, the offline maps of scheduling and control are embedded in the overall MIDO problem in the gPROMS environment. The reactor configuration with volumes $V_1 = 0.44m^3$ and $V_2 =$

$0.92m^3$ minimizes the total annual cost accounting for the capital and operating costs. Note that the optimizer selects one large reactor and one small reactor that delivers (i) uninterrupted production of one of the products depending on their unit storage prices and demand rates throughout the horizon, and (ii) fast transitions for alternating production of the remaining products, respectively.

4.3.3 Single residential heat and power unit

This case study presents an application of a combined heat and power generation system (CHP) on a residential scale. In our previous work [88], we developed design dependent explicit controllers to simultaneously optimize the design and control decisions in a MIDO formulation. In this study, we extend this approach by taking into account the external factors that affect the desired level of operation, i.e. the fluctuations in the heat and power demand rates, and changing market prices for the electricity and fuel. We consider a residential district with 10 units, all of which are supplied hot water for heating purposes and electricity from a single CHP unit. The hot water can be stored in a buffer tank if the produced heat content exceeds the demand rate. Additional electricity can be supplied from the central grid if the CHP unit falls short, and a supplementary boiler is assumed to be available at all times to provide more heat content. Excess electricity produced from the CHP unit can be sold to the central grid for revenue, and excess hot water can be disposed of at an expense. Note that the rapidly changing electricity prices in day time and night time has a significant economic impact on the operation of a CHP unit. For instance, it may be more profitable to operate the CHP unit at a higher capacity during the day time because of the increased cost of electricity purchase, and at a lower capacity at the night time when the cost decreases. Therefore, determining the most cost effective operation can be achieved by taking into account the fluctuation in the prices, demands rates, as well as the dynamics of the CHP units. A generalized flowsheet of the CHP system with two parallel CHP units is presented in Fig. 4.4. However in this section, we focus on a system with a single CHP system supplying the heat and power to the residential units. Parallel operation of multiple units will be discussed in the subsequent example.

The problem statement is then given as follows:

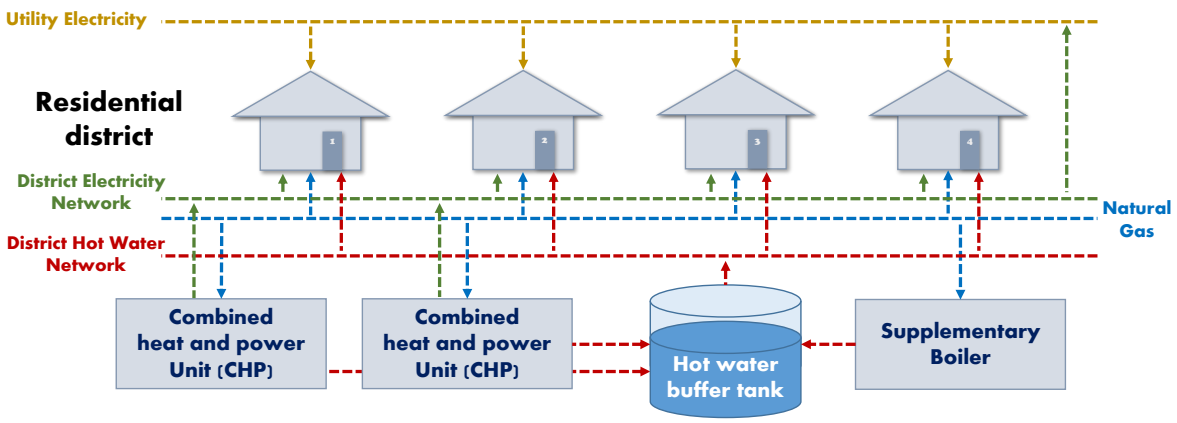


Figure 4.4: CHP example – A generalized flowsheet of the CHP system (Reprinted with permission from [8]).

(i) *Given*: A high-fidelity model of the CHP, a demand scenario for electricity and heat consumption, investment cost of the CHP unit as a function of its size, market prices of fuel and purchasing/selling electricity.

(ii) *Determine*: Internal combustion engine (ICE) size of the CHP, a schedule for the transactions with the grid and fuel purchases, operating level of the CHP.

(iii) *Objective*: Minimize the sum of operating and capital costs.

The size of the ICE directly affects the process time of the system, and thus the responsiveness of the CHP to fluctuations in the demand rates and market prices. ICEs smaller in size have lower transition time, hence they can deliver fast responses to changes in the operating set points. On the other hand, larger ICEs can supply more power and heat to the residential units when the demand rates are high. The trade-off between the responsiveness and the capacity of the CHP is addressed by integrating a design dependent scheduler and controller in the design optimization problem.

High-fidelity dynamic model: There are two main components taken into account in the CHP model, (i) a natural gas powered ICE to produce electrical power, and (ii) a cooling system that recovers the excess heat content of the ICE. We also include the dynamics of the throttle valve that manipulates the inlet air mass flow rate, and the intake manifold that distributes the air into the ICE

cylinders. For the detailed mathematical model, the reader is referred to Diangelakis et al. (2014) [198].

Model approximation: The original high fidelity model is a DAE system with 364 algebraic and 15 differential relations in the continuous domain. The complexity of the overall system was previously addressed by decomposition into two approximate models, namely a power production subsystem and a heat recovery subsystem [199, 88]. The former operating mode gives the relation between the throttle valve opening and the power output of the CHP, while the latter is used to estimate the water temperature at the outlet as a function of the power output and the water flow rate into the heat recovery system. Equation 4.3 presents the identified state-space model for the power production subsystem.

$$\begin{aligned} x_{t_c+1} &= 0.9799 x_{t_c} + 0.0006 u_{t_c} + 6.516 V \\ y_{t_c} &= 7.839 x_{t_c} \end{aligned} \tag{4.3}$$

where x is the identified state, u is the throttle valve opening, V is the volume of the ICE, y is the electrical power generated by the CHP.

The heat recovery subsystem is an explicit function of the output of the power production subsystem and is given in Eq. 4.4.

$$\begin{aligned} x_{t_c+1} &= \begin{bmatrix} 0.997 & 0.103 & -0.003 \\ -0.002 & 0.940 & 0.116 \\ -0.058 & -0.056 & 0.179 \end{bmatrix} x_{t_c} + \begin{bmatrix} -0.008 & 0.001 \\ 0.280 & -0.033 \\ -1.280 & 0.146 \end{bmatrix} u_{t_c} \\ \hat{y}_{t_c} &= \begin{bmatrix} -529.9 & -2.827 & 0.252 \end{bmatrix} x_{t_c} \end{aligned} \tag{4.4}$$

where x is the set of identified states, u are the the power generation level and water flow rate, respectively, and $\hat{y}$ is the prediction of the hot water temperature at the outlet. The discretization time steps of the models presented in Eq. 4.3 and 4.4 are both 0.1s.

Design of the mpMPC: The two subsystems derived in the previous step are operated by a decentralized control policy, which comprises interlinked control strategies for each subsystem.

We define two operational modes for the decentralized control policy defined as follows.

- *Electricity driven mode (Mode 1):* The operating level of the CHP, i.e. the power set point, is determined based on the power demand. Therefore, the throttle valve opening is manipulated primarily to satisfy the demand on electricity. The operating level projected by the electricity generation subsystem is treated as a measured disturbance by the heat recovery subsystem, hence the produced heat is a function of the power output of the CHP. The heat production level of a standalone CHP unit can be insufficient to satisfy the heat demand at a given time, requiring the use of the supplementary boiler. It is also possible that the produced heat content exceeds the heat demand, in which case the hot water is stored in a buffer tank.

- *Heat production driven mode (Mode 2):* The operating level of the CHP is determined based on the heat demand. Tracking a water temperature set point at $70\,^{\circ}\mathrm{C}$, heat recovery subsystem (i) determines an operating level set point to ensure sufficient heat production by the power production subsystem, and (ii) manipulates the cooling water flow rate to recover enough heat to satisfy the demand. Analogous to mode 1, the power production level may not match the electricity demand. In case of insufficient power, additional electricity is purchased from the central grid, and excess electricity is sold back to the grid for revenue.

The reader is referred to Diangelakis et al. [199, 88, 131] for more details on the operating modes of the system and a quantified evaluation of the decentralized control policy.

Note that changing the operating modes creates an offset between the new set point and the current output of the system. This offset has economical consequences on the operation and dictates the quantity of electricity purchases/sales, usage of the buffer tank and the supplementary boiler. These economical aspects are addressed and mitigated in the following steps.

Closed loop validation: The design dependent decentralized control policy is validated against the high fidelity model under a range of different design and scheduling decisions. Figure 4.5 shows a closed loop simulation of a CHP with $V = 1500cc$ operated with mode 1 only. The power set point is subject to random changes throughout the operation.

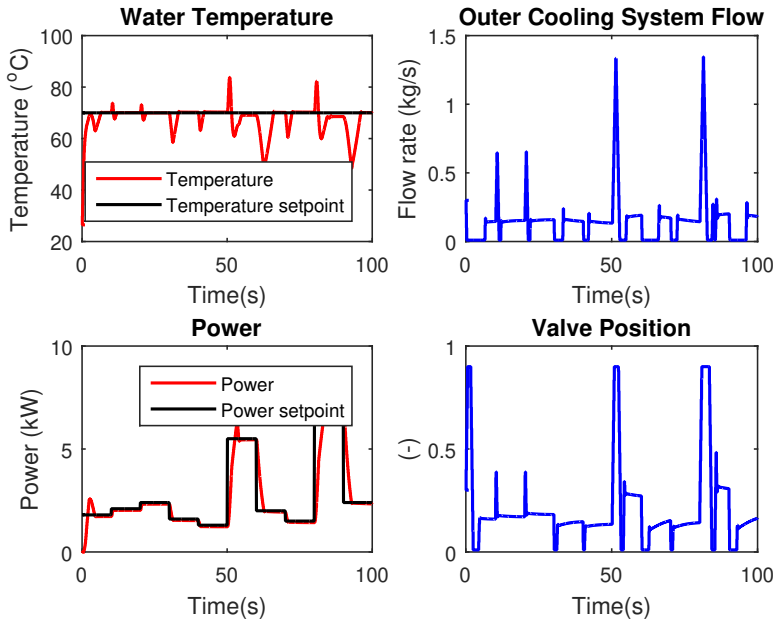


Figure 4.5: CHP example – Closed loop simulation of a CHP unit with $V = 1500cc$, operated with mode 1 (Reprinted with permission from [8]).

Similarly, closed loop simulation on a larger CHP ($V = 5000cc$) is demonstrated in Figure 4.6. Note that due to operating mode 2, the power set point is subject to changes dictated by the heat recovery subsystem.

High fidelity model with the mpMPC embedded: The explicit form of the decentralized control policy is implemented in the original high fidelity model.

Model approximation: The closed loop high fidelity model is used to develop an approximate model for the scheduler via the MATLAB System Identification Toolbox. The identified model establishes a relation between the power production and heat storage levels, and the change in the power production set point, as presented in Eq. 4.5.

$$\begin{bmatrix} E_{t_s+1} \\ B_{t_s+1} \end{bmatrix} = \begin{bmatrix} 0.999 & 0 \\ 37.9 & 0.955 \end{bmatrix} \begin{bmatrix} E_{t_s} \\ B_{t_s} \end{bmatrix} \begin{bmatrix} 99.5 & 0 & 0 \\ 0 & 11.2 & -11.2 \end{bmatrix} \begin{bmatrix} R_{t_s} \\ Q_{t_s} \\ D_{t_s} \end{bmatrix} + \begin{bmatrix} 0 \\ -11.2 \end{bmatrix} \zeta_{t_s}^h \tag{4.5}$$

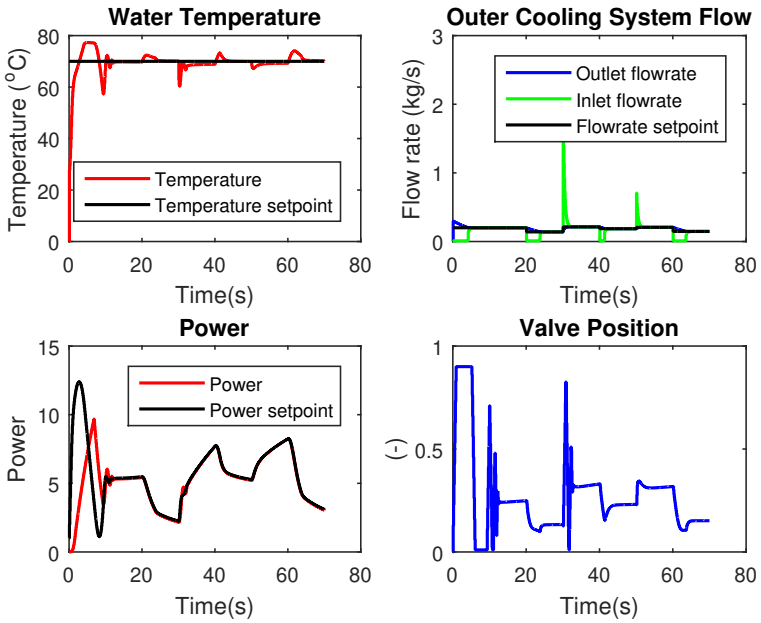


Figure 4.6: CHP example – Closed loop simulation of a CHP unit with $V = 5000cc$, operated with mode 2 (Reprinted with permission from [8]).

where E is the energy production level, B is the heat storage level, R is the change in the power production set point, Q is the additional heat supplied from the boiler, D is the disposed heat, ζ^h is the heat demand, and the time step of the model is 10 s. We also use an overall energy balance for the relation between the power production, power demand, and electricity purchases from the central grid, presented in Eq. 4.6.

$$P_{t_s} + E_{t_s} = \zeta^p_{t_s} + W_{t_s} \tag{4.6}$$

where P is the electricity purchase, ζ^p is the power demand, and W is the excess electricity sold back to the grid.

Design of the scheduler: The objective of the schedule is to minimize the operating costs, including energy production, energy purchases and sales, and inventory costs, as given in Eq. 4.7.

$$\sum_{t=1}^{N_s} \beta E_{t_s} + \psi_{t_s} P_{t_s} - \nu_{t_s} W_{t_s} + \xi_{t_s} Q_{t_s} + \omega_{t_s} D_{t_s} + \gamma B_{t_s} \tag{4.7}$$

where the Greek letters denote the corresponding cost parameters. Note that the CHP unit is assumed to be operational throughout the scheduling horizon. Hence, on/off switching costs are excluded in the objective function. This assumption will be relaxed in Section 4.3.4 where we discuss a parallel operation of multiple CHP units. The objective function is subject to the approximate CHP model derived in Eqs. 4.5 and 4.6, as well as the lower and upper bounds on the optimization variables.

The power production capacity of the CHP unit is a function of the ICE size (i.e. $\overline{E} = \overline{E}(V)$). The schedule treats this design variable as a bounded parameter along with the initial conditions of the system, power and heat demands, unit cost of purchasing fuel and power, and unit revenue of selling power, as listed in Eq. 4.8.

$$\theta = [V, E_{t_s}, B_{t_s}, \zeta^h_{t_s}\zeta^p_{t_s}, \beta_{t_s}, \psi_{t_s}, \nu_{t_s}, \xi_{t_s}, \omega_{t_s}, \gamma_{t_s}] \tag{4.8}$$

Design of the surrogate model: Equations 4.5 and 4.6 are resampled in the time steps of the controller, and substituted in the surrogate model formulation presented in Eq. 3.7. The resampled state space matrices are given in Appendix 4.9.

$$\begin{bmatrix} E_{t_{sm}+1} \\ B_{t_{sm}+1} \end{bmatrix} = \begin{bmatrix} 1.0000 & 0 \\ 0.3880 & 0.9995 \end{bmatrix} \begin{bmatrix} E_{t_{sm}} \\ B_{t_{sm}} \end{bmatrix} + \begin{bmatrix} 0.9954 & 0 & 0 \\ -19.2613 & 0.1143 & -0.1143 \end{bmatrix} \begin{bmatrix} R_{t_{sm}} \\ Q_{t_{sm}} \\ D_{t_{sm}} \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ -0.1143 & 0.0001 \end{bmatrix} \begin{bmatrix} \delta^h_{t_{sm}} \\ V \end{bmatrix} \tag{4.9}$$

Closed loop validation: The integrated scheduling and control scheme is validated against an extensive set of design options and demand profiles. Figure 4.7 shows a snapshot of a closed loop simulation of a CHP unit with a volume $V = 5000cc$. Note that the power set point throughout the operation is determined by the offline schedule, and translated into the time steps of the controller

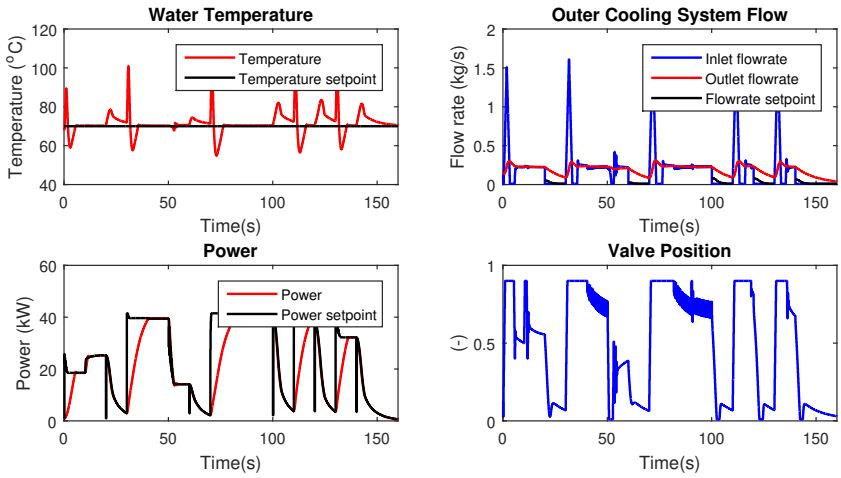


Figure 4.7: CHP example – Closed-loop validation of the integrated scheduling and control scheme on a CHP with $V = 5000cc$ (Reprinted with permission from [8]).

by the surrogate model.

Design optimization: We formulate a MIDO problem in the gPROMS environment using the high fidelity model, the explicit design dependent relations for the scheduler, surrogate model, and the controller. The capital investment cost is assumed to be a linear function of V, and is given in Appendix A.2. A CHP unit with an ICE volume of $V = 1710cc$ yields the scheduling and control strategies that minimizes the total annualized cost that includes the capital and operating costs.

4.3.4 Two residential heat and power units in parallel

The single CHP case study presented in Section 4.3.4 is extended to include two CHP units operating in parallel. We generalize the scheduling formulation to account for multiple CHP units, and showcase the proposed algorithm on a system with two units. We also include the dynamics stemming from switching on/off the units, and their impact on the operational optimization.

Design of the scheduler: Evidently, multiple CHP units have a greater capacity to supply heat and power compared to a single unit. However, the total production rate of multiple units can exceed the demand rates significantly even when they are operated at their lowest capacities. In other words, operating one CHP unit standalone can be more cost effective than operating two CHPs simultaneously at low demand rates. Therefore, we include the start-up and shut-down

dynamics in the schedule to account for the trade-off between switching on/off the operation and maintaining the operating status of a unit.

The cost of switching on/off is described by Eq. 4.10.

$$\sum_{i=1}^{N_{CHP}} \sum_{t=1}^{N_s} \phi_i S_{i,t_s} + \pi_i F_{i,t_s} \tag{4.10}$$

where N_{CHP} is the number of CHP units, S_{i,t_s} and F_{i,t_s} are binary variables that indicate the start-up and shut-down status, and ϕ_i and π_i are their unit costs, respectively. The impact of the switching status variables is incorporated in the schedule by introducing lifting-state variables, $\tilde{S}_{i,t_s,n}$ and $\tilde{F}_{i,t_s,n}$, as presented in Eq. 4.11.

$$\begin{aligned} \tilde{S}_{i,t_s+1,n} &= \tilde{S}_{i,t_s,n-1}, \quad \tilde{S}_{i,t_s,n=0} = S_{i,t_s} \\ \tilde{F}_{i,t_s+1,n} &= \tilde{F}_{i,t_s,n-1}, \quad \tilde{F}_{i,t_s,n=0} = F_{i,t_s} \end{aligned} \tag{4.11}$$

The state lifting-variables determine the operating status of the CHP units as described in Eq. 4.12.

$$\begin{aligned} S_{i,t_s} - F_{i,t_s} &= X_{i,t_s} - X_{i,t_s-1} \\ X_{i,t_s} &\geq \sum_{n=0}^{\delta_i^{up}} \tilde{S}_{i,t_s,n} \\ 1 - X_{i,t_s} &\geq \sum_{n=0}^{\delta_i^{dn}} \tilde{F}_{i,t_s,n} \end{aligned} \tag{4.12}$$

where X_{i,t_s} is a binary variable that indicate the operating status, δ_i^{up} and δ_i^{dn} are the start-up and shut-down times of the i^{th} CHP unit. The interested reader is referred to Subramanian et al. (2012) [184] for more details on scheduling with lifting-state variables, and to Kopanos et al. (2014) [187] for an application of reactive scheduling using lifting-state variables on a CHP system.

The cost function given in Eq. 4.7 is generalized to encapsulate the operating cost of multiple CHP units, as presented in Eq. 4.13.

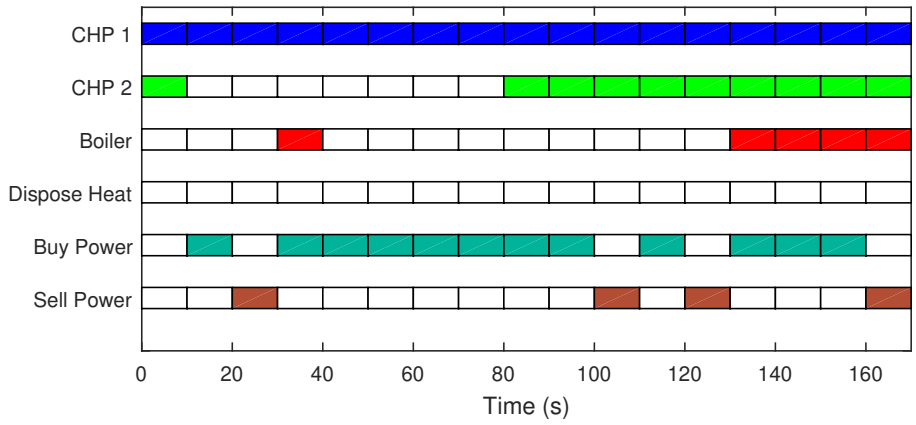


Figure 4.8: Two CHPs in parallel - Closed loop simulation of the generalized scheduling scheme in two CHP units operating in parallel. The volumes of the ICE are $V_1 = 1500cc$ and $V_2 = 4500cc$, respectively (Reprinted with permission from [8]).

$$\sum_{i=1}^{N_{CHP}} \sum_{t=1}^{N_s} \beta_{t_s} E_{i,t_s} + \sum_{t=1}^{N_s} \psi_{t_s} P_{t_s} - \nu_{t_s} W_{t_s} + \xi_{t_s} Q_{t_s} + \omega_{t_s} D_{t_s} + \gamma B_{t_s} \tag{4.13}$$

The objective function of the schedule comprises the operating and purchasing costs described by Eq. 4.13 and the switching costs given in Eq. 4.12.

Closed loop validation: The developed scheduling strategy is implemented on the high fidelity model and operated in tandem with the offline controller. Figure 4.8 shows a snapshot of the scheduling level decisions of an operation with two CHP units with ICE volumes $V_1 = 1500cc$ and $V_2 = 4500cc$, under a rapidly escalating demand profile given in Figure 4.9. The following are some observations and remarks on the closed loop performance of the developed scheduling and control strategies.

4.4 Conclusions

In this chapter, a novel, process agnostic framework was introduced to integrate the design, scheduling, and control problems based on a single high fidelity model. Using multiparametric programming, we derived offline piecewise strategies for (i) a control scheme as a funtion of the design and scheduling decisions, (ii) a scheduling scheme as a function of design, and aware of

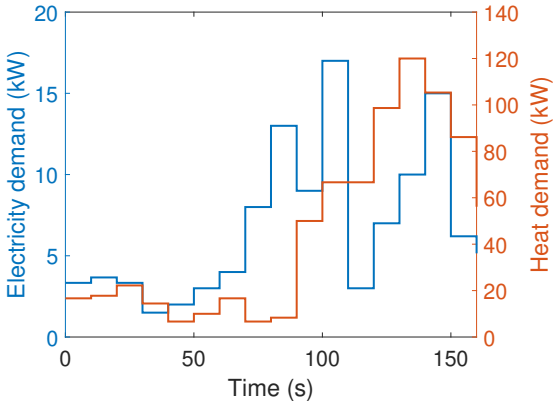


Figure 4.9: Two CHPs in parallel – Snapshot of the electricity and heat demand profiles. Note the steep increase in demand in short notice (Reprinted with permission from [8]).

the closed-loop dynamics through a surrogate model formulation. The offline maps of strategies allowed for a direct implementation in a MIDO formulation for design optimization. The proposed framework was able to determine the process design that guarantees the operability of the system under a range of bounded process and market uncertainties by simultaneously considering the optimal scheduling and control strategies used in closed-loop implementation.

Postulating all layers of decisions as optimization problems has specific benefits to tailor each individual problem based on the needs of the system of interest. This advantage was illustrated by using soft constraints to satisfy product purity in the CSTR examples, and by using a decentralized control structure in the CHP examples. Note that the framework was applied on both problems without appealing to further modifications.

The major bottleneck of the proposed framework is employing approximate models in the control and scheduling levels. Although the confidence on the models were increased by using well-established and previously proposed error metrics [200], the approximation creates a mismatch between the real process dynamics and the decision making optimization problems. Future work will focus on incorporating robust counterparts of the scheduling and control problems to account for the mismatch. However, robust multiparametric receding horizon policies result in an explosion in the number of critical regions in the parametric solution space. This increase is also

a natural result of highly dynamic processes, such as batch mode operations. Therefore in Chapter 5, we propose a computationally efficient modeling technique and optimization framework to integrate the optimal operational strategies in an MIDO formulation for batch processes.

5. INTEGRATED PROCESS DESIGN OPERATIONAL OPTIMIZATION OF BATCH PROCESSES[1]

Batch processing has been the predominant choice of operation mode to manufacture high value specialty chemicals due to its inherent flexibility to satisfy volatile customer requirements. Short term scheduling in batch processing is a key factor towards delivering the targeted production requirements by the end of a predetermined horizon, as the scheduling implementation can often dictate the profitability of the entire process especially if a high number of products is to be manufactured in a limited number of multipurpose equipment [201, 202].

A scheduling problem comprises a variety of decisions such as resource allocation, task sequencing, and task timing. State-Task Network (STN) [66] and Resource-Task Network (RTN) [203] are two of the most widely used scheduling techniques that provide a systematic modeling framework and solution strategy for these decisions through mixed-integer linear programming (MILP). STN/RTN adopt a recipe based scheduling approach, where the batch sizes and processing times are assumed to be fixed. Continuous-time scheduling approaches improve upon this limitation by using linearized relations for the batch sizes and processing times [202, 204, 205, 206, 207, 208, 209]. However, the optimality and even the feasibility of the schedule is susceptible to internal and external influences such as different initial conditions, known/unknown process disturbances, and fluctuations in utility and raw material prices. Utilizing static transition tables that comprise processing times or time constants is a common, albeit ad-hoc modeling representation that poses challenges to generalize for all possible cases due to the lack of an in depth understanding of the process dynamics [94].

Model based approaches that integrate scheduling decisions with faster time scale decisions are shown to be promising to account for the dynamic characteristics of the process [1, 194, 9, 210]. Bhatia and Biegler (1996) [92] have proposed one of the first significant contributions to simultane-

[1]Portions of this chapter have been submitted for publication as Burnak, B., Katz, J., Pistikopoulos, E.N., Integrated process design, scheduling, and model predictive control of batch processes with closed-loop implementation, *in review*.

ously address the process design, scheduling, and optimal control of a multipurpose batch process in a dynamic optimization formulation. The authors formulated a dynamic model for the batch process in continuous time domain, which was discretized into a finite dimensional nonlinear programming problem (NLP) and solved using orthogonal collocation on finite elements. Biegler and co-workers extended the use of dynamic models in an integrated formulation with more comprehensive and practically relevant scheduling schemes, state equipment networks (SEN) [169] and RTN [170]. Chu and You (2014) [211] have proposed a surrogate modeling based approach for the integration planning, scheduling, and open loop dynamic optimization for processes with fixed batch sizes. More recently, Valdez-Navarro and Ricardez-Sandoval (2019) [212] have addressed the integrated scheduling and control problem via the STN framework and a back-off algorithm to handle process uncertainties. Although these approaches have been demonstrated to capture the key interactions between the site level and unit level process decisions, they are merely intended to be used in the offline phase of decision making. In other words, such open loop optimization approaches neglect the behavior of the feedback controller, which fundamentally changes the dynamics of the process. Earlier studies by Soroush and Kravaris (1993) [134, 133] accounted for the PID type state feedback controllers by incorporating their explicit control laws in a dynamic optimization formulation. However, more advanced control strategies such as constrained Model Predictive Control (MPC) have implicit forms, where the optimal control actions is only available after solving an optimization problem at every step in a rolling horizon manner. Zhuge and Ierapetritou (2014) [177] developed multi-parametric MPCs (mpMPC) to be incorporated in an integrated scheduling and control formulation. However, the proposed approach utilizes an event point based scheduling formulation with variable discretization steps, which creates a mismatch with the fixed step size of the state space model used in the mpMPC. Rossi et al. (2017) [213] have proposed a two phase architecture for the integrated problem where the first phase solves a conventional scheduling problem offline and the second phase comprises the online implementation of a modified nonlinear MPC (NMPC).

In Chapter 4, we discussed the use of multi-parametric programming to derive explicit expres-

sions to be integrated in an MIDO formulation with a focus on continuous processes. However, batch processes are inherently transient and track time varying input-output trajectories. As a consequence, the explicit solutions to optimal operational decisions are usually large in size, which renders the problem computationally expensive. Therefore in this chapter, we introduce an efficient modeling and optimization framework to integrate the design and operational decisions in batch processing, while accounting for the increasing size in explicit solutions. Accounting for the MPC dynamics in the integrated problem allows for the derivation of closed-loop optimal trajectories that are attainable by the advanced control scheme, thereby offering certificates of operability for the closed-loop implementation. We utilize the SEN framework for the scheduling problem due to its suitability for the integration with the optimal control problem [169]. Moreover, we introduce a methodology to exponentially reduce the number of binary variables for embedding the piecewise affine partitions derived from the multi-parametric solution of the MPC based on the base-2 numeral system.

The remainder of the chapter is organized as follows. Section 5.1 defines the integrated problem and the types of decisions that are considered in this chapter. In Section 5.2, we present a mathematical formulation of the complete integrated problem, the methodology to derive explicit MPC strategies that govern the system of interest, the essential components of the SEN framework, and the methodology to embed the explicit MPC solution into the resulting mixed-integer dynamic optimization formulation. Finally, we showcase the proposed approach with two batch process examples in Section 5.3.

5.1 Problem Statement

We consider a multipurpose batch process where the products are allowed to follow different routes through the plant at different times [214]. The objective of these batch plants may vary depending on the application, such as minimizing the cost, minimizing the makespan, or maximizing the yield of a specific product. The goal of this work is to present a unified theory and framework to determine simultaneously the following four levels of operational decisions, while delivering the target objective. Therefore, the problem statement is illustrated in Fig. 5.1 and outlined as follows.

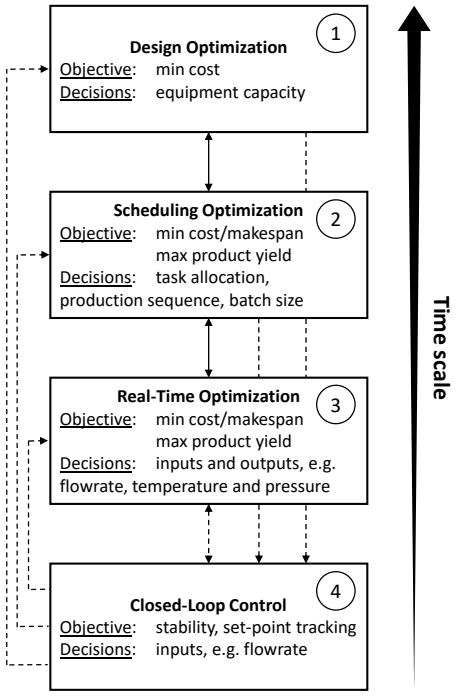


Figure 5.1: A schematic of the objectives and decisions in each multi-scale level. The dashed lines denote the particular contribution of this study.

Given. First principle dynamics to manufacture the desired products (preprocessing, reaction, separation), any physical limitations regarding the product quality and process safety, unit capital and operating costs, and the range of demands on products.

Determine.

i. Process design decisions: Dictates the capacity of the processing units.

ii. Process scheduling decisions: Includes task allocation, production span or cycle, production sequence, and batch sizes.

iii. Real-time optimization decisions: Input and output trajectories that are transmitted to the regulatory controller.

iv. Closed-loop control decisions: Model predictive control strategy that governs the process through the control instruments.

Objective. Minimize cost, minimize, makespan, maximize yield, etc.

Figure 5.1 illustrates the objectives and decisions associated with each layer considered in this chapter in their hierarchical time-scale order. In Table 5.1, we present some of the notable studies that explore the connectivities between these layers with their scopes and their significant contributions in the field. The main contribution of this chapter, marked with dashed lines in Fig. 5.1, exploits synergistic interactions between advanced closed-loop control strategies and longer term decisions to provide certificates of operability at each individual level. In Section 5.2, we first provide a conceptual mathematical representation of the integrated problem, followed by a framework to develop an offline MPC strategy for a batch process, characteristic equations in the SEN framework, and an exact modeling technique to integrate the advanced controller in a mixed-integer dynamic optimization (MIDO) formulation via logical disjunctions.

5.2 Problem Formulation

A generalized mathematical form of the integrated design and operation optimization problem has been already provided in 1.11. Therefore in this chapter, we formulate a bilevel MIDO problem, given by Eq. 5.1, where the MPC dynamics are accounted for as the "follower" in the integrated problem. In Eq. 5.1, the integrated problem features (i) design decisions as time-invariant variables, (ii) scheduling decisions based on a SEN representation, (iii) open loop optimal control profiles used as set points for the feedback controller, and (iv) closed-loop MPC dynamics embedded via multi-parametric programming.

Table 5.1: An indicative list of the notable studies with their scopes.

Authors	Design decisions	Scheduling decisions	Real-time optimization	Closed-loop control	Significant contribution
Soroush and Kravaris (1993) [134, 133]	✓	✗	✓	✓	Incorporated PID control in a DO formulation with notions of feasibility, flexibility, controllability, and safety.
Bhatia and Biegler (1996) [92]	✓	✓	✓	✗	Infinite dimensional DO is solved via ortogonal collocation on finite elements.
Nie et al. (2012) [169]	✓	✓	✓	✗	Use of SEN for scheduling decisions.
Zhuge and Ierapetritou (2012) [101]*	✗	✓	✓	✓	Closed-loop implementation with an iterative approach.
Chu and You (2013) [215]	✗	✓	✓	✗	Stochastic programming and Generalized Benders Decomposition based approach.
Zhuge and Ierapetritou (2014) [177]	✗	✓	✓	✓	Closed-loop strategies accounted for via multi-parametric programming.
Nie et al. (2014) [170]	✓	✓	✓	✗	Use of RTN for scheduling decisions.
Du et al. (2015) [103]*	✗	✓	✓	✓	Use of low-dimensional scale bridging models.
Baldea et al. (2015) [104]*	✗	✓	✓	✓	Use of low-dimensional scale bridging models with model-based control.
Valdez-Navarro and Ricardez-Sandoval (2019) [212]	✗	✓	✓	✗	Implemented back-off approach with Monte Carlo simulations to account for uncertainty.

*Cyclic continuous process – included in the list due the applicability of the approach to batch processes.

$$
\begin{aligned}
\min_{u(t),s(t),des} \quad & \int_0^\tau C(x(t),y(t),u(t),s(t),des,d(t))dt \\
s.t. \quad & \dot{x}(t) = f(x(t),u(t),s(t),des,d(t)) \\
& \underline{y} \leq y(t) = g(x(t),u(t),s(t),des,d(t)) \leq \overline{y} \\
& \underline{u} \leq u(t) \leq \overline{u}, \quad \underline{s} \leq s(t) \leq \overline{s} \\
& \underline{x} \leq x(t) \leq \overline{x}, \quad \underline{des} \leq des \leq \overline{des}, \quad \underline{d} \leq d(t) \leq \overline{d} \\
& x(0) = x_0 \\
\min_{u_k} \quad & \sum_{k \in OH} \|x_k\|^2_{Q_k} + \|y_k - y_k^{sp}\|^2_{QR_k} + \sum_{k \in CH} \|u_k - u_k^{sp}\|^2_{R_k} + \|\Delta u_k\|^2_{R1_k} \\
s.t. \quad & x_{k+1} = Ax_k + Bu_k + Cd_k \\
& y_k = Dx_k + Eu_k + Fd_k \\
& \underline{x} \leq x_k \leq \overline{x}, \quad \underline{y} \leq y_k \leq \overline{y}, \quad \underline{d} \leq d_k \leq \overline{d} \\
& \underline{u} \leq u_k \leq \overline{u}, \quad \underline{\Delta u} \leq \Delta u_k \leq \Delta \overline{u}
\end{aligned}
\tag{5.1}
$$

where the follower problem is a standard MPC formulation. In Eq. 5.1, $\|\cdot\|_\Psi$ denotes the weighted vector norm with a weight matrix Ψ, OH is the index set of the output horizon, CH is the index set of the control horizon, Q_k, QR_k, R_k, $R1_k$ are the weight matrices of the states, process outputs, process inputs, and input deviations, respectively, and sp denotes the set point. Note that the MPC formulation introduces an additional connectivity between the inputs and outputs of the system that is neglected in Eq. 1.11, which inherently changes the process dynamics. Therefore, inclusion of the MPC dynamics is paramount to achieve realizable optimal trajectories.

In Chapter 4, we presented a theory and framework for the integration of MPC in design optimization [88], and the incorporation of scheduling decisions via multi-parametric programming [8]. Although this theory is applicable to batch processes in principle, the practical implementation becomes a challenging task as the control horizon and the number of manipulated variables in the MPC scheme increase, which is frequently encountered in batch processes. Increasing the

number of decision variables in the mpMPC formulation results in an exponential increase in the number of critical regions, all of which contain the optimal control law to be used based on the online state measurements. In the integrated formulation presented in Eq. 5.1, the critical regions are embedded via a big-M or convex hull formulation, requiring the use of a binary variable for each critical region throughout the optimization horizon.

In the following discussions, we detail the constituents of the MIDO problem, i.e. (i) SEN formulation with its common assumptions, and (ii) the integration of the mpMPC in the SEN and dynamic optimization formulation. Developing the mpMPC that governs the process has been already discussed in Chapters 2 and 3. The complete formulation of the MIDO is given in Appendix A.3 in a generic form.

5.2.1 Scheduling using the state equipment network

In this section, we discuss the scheduling optimization via the SEN representation outlined in Panel 2 of Fig. 5.1. Nie et al. (2012) [169] discussed the suitability of the SEN framework for the integration of the scheduling decisions into a dynamic optimization formulation via generalized disjunctive programming. The authors adopted the unit specific event-based continuous time representation, where the scheduling horizon is divided into a finite number of event slots for each unit. Although this approach is practical for open loop dynamic optimization, it is a challenging task to apply on a process governed by an MPC scheme due to two reasons. First, the bilevel nature of the integrated problem poses a modeling challenge, which will be discussed in Section 5.2.2. Second, the MPC strategy acts on the system in a rolling horizon fashion, where the optimal control action is updated in discrete time intervals, creating a mismatch with the continuous optimal trajectory proposed by Nie et al. (2012) [169]. Since the discretization steps of the MPC is fixed by design prior to operation, we use evenly distributed discrete time intervals in the SEN framework to integrate the dynamic model and the mpMPC via logical disjunctions. In this part, we discuss the essential constraints and objective functions that can be used in the discrete-time SEN framework.

Assignment constraints. We define a set if binary variables $y_{j,s,t}$ that denote operating state s of an equipment j in time slot t. $y_{j,s,t}$ is equal to 1 if equipment j is occupied by state s in time slot t, and 0 if otherwise. Therefore, we use Eq. 5.2 to dictate the exclusivity of states in an equipment throughout the scheduling time horizon.

$$\sum_{s \in \mathcal{S}} y_{j,s,t} \leq 1, \quad \forall j \in \mathcal{J}, \forall t \in \mathcal{T} \tag{5.2}$$

Similarly, one task can only be executed in one equipment, as given by Eq. 5.3.

$$\sum_{j \in \mathcal{J}} y_{j,s,t} \leq 1, \quad \forall s \in \mathcal{S}, \forall t \in \mathcal{T} \tag{5.3}$$

Continuity constraints. After a task is assigned to an equipment, it has to continue the process in the same equipment.

$$y_{j,s,t+1} \leq y_{j,s,t}, \quad \forall j \in \mathcal{J}, \forall s \in \mathcal{S}, \forall t \in \mathcal{T}, t \neq t_f \tag{5.4}$$

where t_f is the final scheduling time step.

Material balance. At each discretization point, we construct the material balance for every component c to determine their availability, as presented in Eq. 5.5.

$$E_{c,t} = E_{c,t-1} + \sum_{j \in \mathcal{J}} \Delta E_{j,c,t}, \quad \forall c \in \mathcal{C}, \forall t \in \mathcal{T}, t > 0 \tag{5.5}$$

where $E_{c,t}$ denotes the amount of excess material of component c at time t, and $\Delta E_{j,c,t}$ is the generation or consumption term, dictated by the reaction kinetics in the high-fidelity model.

Capacity constraints. The vessel sizes limit the amount of material that can be processed in every task.

$$\sum_{s \in \mathcal{S}} y_{j,s,t} \underline{V} \leq V_{j,t} \leq \sum_{s \in \mathcal{S}} y_{j,s,t} \overline{V}, \quad \forall s \in \mathcal{S}, \forall t \in \mathcal{T} \tag{5.6}$$

where $V_{j,t}$ is a set of continuous variables that describe the volume occupied in equipment j. Note that it is possible to enforce similar constraints on the excess material $E_{c,t}$. However, we assume unlimited intermediate storage (UIS) and neglect such constraints for simplicity.

Quality constraints. These constraints are included to enforce certain quality metrics, such as product purity or target demand, at the end of the batch.

$$\begin{aligned} & x_s^* \leq x_{s,t+1} + \mathcal{M}(w_{s,t}), \quad \forall s \in \mathcal{S}^*, t = 0 \\ & x_s^* \leq x_{s,t+1} + \mathcal{M}\big(1 - (w_{s,t} - w_{s,t+1})\big), \quad \forall s \in \mathcal{S}^*, \forall t \in \mathcal{T}, 0 \leq t \leq t_f \\ & x_s^* \leq x_{s,t+1} + \mathcal{M}\big(1 - w_{s,t}\big), \quad \forall s \in \mathcal{S}^*, t = t_f \end{aligned} \tag{5.7}$$

where superscript "*" denotes the target states, $\mathcal{M}$ is a sufficiently large number for the big-M formulation, and $w_{s,t}$ is defined as a set of linking variables between the scheduling model and the dynamic high fidelity model. The linking variables $w_{s,t}$ are a set of Boolean variables that are enforced to have a "true" value if the task is still in progress via Eq. 5.7, and "false" if otherwise. These variables are linked to the scheduling model as presented by Eq. 5.8.

$$(t+1)w_{s,t} \leq \sum_{j' \in \mathcal{J}} \sum_{t' \in \mathcal{T}} y_{j',s,t'}, \quad \forall s \in \mathcal{S}, \forall t \in \mathcal{T} \tag{5.8}$$

Sequence constraints. In the case that one state should take place only after the completion of another task (e.g. precursors), the priority can be dictated by Eq. 5.9.

$$y_{j,s^+,t} \leq \sum_{t'=0}^{t} y_{j,s^-,t'}, \quad \forall j \in \mathcal{J}, \forall s^- \in \mathcal{S}^-, \forall s^+ \in \mathcal{S}^+, \forall t \in \mathcal{T} \tag{5.9}$$

where superscript "$^-$" denotes the states that should be scheduled earlier than the states labeled by the superscript "$^+$".

Objective functions. Here, we will present two most commonly used objectives in a process schedule, although they can be diversified and tailored to serve different purposes. For makespan minimization, a set of Boolean variables z_t is defined to indicate if the overall process is still in

progress.

$$y_{j,s,t} \leq z_t, \quad \forall j \in \mathcal{J}, \forall s \in \mathcal{S} \tag{5.10}$$

Then, the makespan of one batch cycle can be minimized by minimizing the sum of z_t, as presented by Eq. 5.11.

$$\sum_{t \in \mathcal{T}} z_t \tag{5.11}$$

Similarly, cost minimization is one of the most common objectives encountered in processes schedules, and can be expressed by Eq. 5.12.

$$\sum_{t \in \mathcal{T}} C^u u_t + \sum_{t \in \mathcal{T}} C^t z_t \tag{5.12}$$

5.2.2 Integrating mpMPC in the MIDO

In Chapters 2 and 3, we introduced a systematic procedure to develop MPC schemes based on a high fidelity model, and to derive the explicit form of the optimal control law, given by Eq. 2.4. In this section, we will introduce an efficient methodology to integrate the optimal control law with significantly less binary variables, which is previously outlined in Fig. 5.1 with the dashed lines.

The optimal control law is expressed by a piecewise affine expression, and has two components, namely (i) a set of affine functions that are optimal for the polytopic space CR_n (Eq. 2.4a), and (ii) a set of polytopes that define the space that bound the corresponding affine expression (Eq. 2.4b). Equation 2.4a can be reformulated by using the two main relaxation schemes, namely big-M reformulation and convex hull formulation. These relaxation schemes can be used to embed the mpMPC to the SEN network via a set of binary variables $y^{CR}_{n,t}$.

$$-\mathcal{M}(1 - y^{CR}_{n,t}) \leq u_t - K_n\theta_t - r_n \leq \mathcal{M}(1 - y^{CR}_{n,t}), \quad \forall n \in NC, \forall t \in \mathcal{T} \tag{5.13a}$$

$$\begin{aligned} &-\mathcal{M}(1 - y^{CR}_{n,t}) \leq u_{n,t} - K_n\theta_t - r_n \leq \mathcal{M}(1 - y^{CR}_{n,t}), \quad \forall n \in NC, \forall t \in \mathcal{T} \\ &u_t = \sum_{n \in NC} u_{n,t}, \quad \forall t \in \mathcal{T} \end{aligned} \tag{5.13b}$$

where Eq. 5.13a represents the big-M reformulation and Eq. 5.13b represents the convex hull reformulation for the optimal control law. We also dictate that at most one critical region can be selected at a given time throughout the scheduling horizon, as given by Eq. 5.14.

$$\sum_{n \in NC} y_{n,t}^{CR} \leq 1, \quad \forall t \in \mathcal{T} \tag{5.14}$$

Selection of the critical region strictly depends on the feasibility of the parameter set $\boldsymbol{\theta}_t$ at time t. Therefore, we can simply relax the disjoint polytopes by Eq. 5.15.

$$A_n^{CR}\theta_t - b_n^{CR} \leq \mathcal{M}(1 - y_{n,t}^{CR}), \quad \forall n \in NC, \forall t \in \mathcal{T} \tag{5.15}$$

Note that both the big-M and convex hull reformulation schemes require a binary variable for every critical region and for each time step throughout the scheduling horizon. Consequently, the computational complexity of the MIDO problem grow exponentially as the number of critical regions of the explicit optimal control law increase. The states of a batch process are inherently time-varying and hence, the MPC scheme of a batch process requires longer output and control horizons, and larger bounds on the variables compared to a typical continuous process. The combinatorial nature of the increased number of variables and constraints of the MPC problem results into an exponential increase in the number of critical regions in its explicit solution. Therefore, employing the big-M and convex hull reformulation techniques become impractical due to the number of the $y_{n,t}^{CR}$ variables, especially for the batch processes. Herein, we present an efficient modeling technique with significantly less binary variables using the base-2 numeral system. The goal of this technique is to represent each critical region in a time step with a unique combination of a set of binary variables, $\bar{y}_{n_2,t}^{CR}$.

Let n_2 denote the n$^{\text{th}}$ critical region in the base-2 numeral system (i.e. $n_2 = n$). We treat the digits of n_2 as an array of binary parameters, denoted by β_{n_2}. Therefore, a generic constraint $g(x)$ can be relaxed with the unique combinations of a set of binary variables y_i as presented by Eq.

5.16.

$$g(x) \leq \mathcal{M}\left(\sum_{i\in\{m|\beta_{n_2,m}=0\}} y_i + \sum_{i\in\{m|\beta_{n_2,m}=1\}} (1-y_i)\right) \tag{5.16}$$

The relaxation scheme presented in Eq. 5.16 reduces the number of required binary variables from n to $\lceil \log_2 n \rceil$, where $\lceil \cdot \rceil$ denotes the ceiling function. Note that if the number of binary combinations is greater than the number of constraints (i.e. $2^{\lceil \log_2 n \rceil} > n$), we need additional constraints to exclude those combinations from the feasible space by integer cuts, as presented by 5.17.

$$\sum_{i\in\{m|\beta_{n_2,m}=1\}} y_i - \sum_{i\in\{m|\beta_{n_2,m}=0\}} y_i \leq |\, m|\beta_{n_2,m}=1 \,| - 1 \tag{5.17}$$

where $|\cdot|$ denotes the cardinality operator. An illustrative example for the use of base-2 numeral system to relax a set of constraints is provided in Appendix A.4.

Using the base-2 numeral system to integrate the explicit MPC solution The base-2 numeral system can be applied to the big-M (5.13a) and convex hull (5.13b) reformulation schemes for the piecewise affine control law as presented by Eqs. 5.18 and 5.19, respectively.

$$-\mathcal{M}\left(\sum_{i\in\{m|\beta_{n_2,m}=0\}} \bar{y}_{i,t}^{CR} + \sum_{i\in\{m|\beta_{n_2,m}=1\}} (1-\bar{y}_{i,t}^{CR})\right) \leq u_t - K_{n_2}\theta_t - r_{n_2}, \quad \forall n_2 \in NC_2, \forall t \in \mathcal{T} \tag{5.18a}$$

$$u_t - K_{n_2}\theta_t - r_{n_2} \leq \mathcal{M}\left(\sum_{i\in\{m|\beta_{n_2,m}=0\}} \bar{y}_{i,t}^{CR} + \sum_{i\in\{m|\beta_{n_2,m}=1\}} (1-\bar{y}_{i,t}^{CR})\right), \quad \forall n_2 \in NC_2, \forall t \in \mathcal{T} \tag{5.18b}$$

$$-\mathcal{M}\left(\sum_{i\in\{m|\beta_{n_2,m}=0\}}\bar{y}_{i,t}^{CR}+\sum_{i\in\{m|\beta_{n_2,m}=1\}}(1-\bar{y}_{i,t}^{CR})\right)\leq u_{n_2,t}-K_{n_2}\theta_t-r_{n_2},\quad\forall n_2\in NC_2,\forall t\in\mathcal{T}$$

(5.19a)

$$u_{n_2,t}-K_{n_2}\theta_t-r_{n_2}\leq\mathcal{M}\left(\sum_{i\in\{m|\beta_{n_2,m}=0\}}\bar{y}_{i,t}^{CR}+\sum_{i\in\{m|\beta_{n_2,m}=1\}}(1-\bar{y}_{i,t}^{CR})\right),\quad\forall n_2\in NC_2,\forall t\in\mathcal{T}$$

(5.19b)

$$u_t=\sum_{n_2\in NC_2}u_{n_2,t},\quad\forall t\in\mathcal{T}\tag{5.19c}$$

Note that we do not enforce Eq. 5.14 in the base-2 numeral system as any feasible combination of the binary variables $\bar{y}_{n_2,t}^{CR}$ yield a unique optimal control law. The feasibility of the control laws in closed loop is analogously satisfied by Eq. 5.20.

$$A_{n_2}^{CR}\theta_t-b_{n_2}^{CR}\leq\mathcal{M}\left(\sum_{i\in\{m|\beta_{n_2,m}=0\}}\bar{y}_{i,t}^{CR}+\sum_{i\in\{m|\beta_{n_2,m}=1\}}(1-\bar{y}_{i,t}^{CR})\right),\quad\forall n_2\in NC_2,\forall t\in\mathcal{T}\tag{5.20}$$

Therefore, using Eqs. 5.18 or 5.13b along with Eq. 5.20 provides an exact integration of the mpMPC into the MIDO formulation. If the number of critical regions n is greater than the number of binary combinations (i.e. $2^{\lceil\log_2 n\rceil}>n$), then we can use Eq. 5.17 by rewriting as follows to eliminate the infeasible combinations.

$$\sum_{i\in\{m|\beta_{n_2,m}=1\}}\bar{y}_{i,t}^{CR}-\sum_{i\in\{m|\beta_{n_2,m}=0\}}\bar{y}_{i,t}^{CR}\leq|\ m|\beta_{n_2,m}=1\ |-1,\quad t\in\mathcal{T}\tag{5.21}$$

5.3 Case Studies

5.3.1 Illustrative example – Single reaction

We consider a reaction that takes place in a batch reactor under nonisothermal conditions. The goal of this case study is to demonstrate the methodology to embed the MPC dynamics in

a dynamic optimization framework. Therefore, the design problem and scheduling via the SEN framework are excluded in this problem for simplicity. Here, we only focus on developing an MPC that manipulates the heat input to track temperature and concentration set points that are determined by the real-time optimization formulation. The stoichiometry of the reaction is given as follows.

$$A \underset{k_{-1}}{\overset{k_1}{\rightleftharpoons}} B \xrightarrow{k_2} C$$

where A is the raw material, B is the desired product, and C is a by-product with negligible monetary value. The reaction setting is selected such that two of the most common challenges in a batch reaction process, namely a reversible reaction and a side reaction, are included. Due to the reverse reaction k_{-1}, complete conversion to product B is thermodynamically infeasible. Furthermore, since the reaction path involves a by-product through an irreversible reaction k_2, the trivial solution of using an infinitely long batch time at the maximum operable temperature is also infeasible to satisfy the product demand. Therefore, a model based dynamic optimization approach should be employed to maximize the desired performance metrics. In this case study, we will demonstrate makespan minimization and yield maximization as performance metrics.

The high-fidelity dynamics of the reaction is developed as a set of DAE and presented as follows. First, we consider the mass and energy balances, as given by Eqs. 5.22a and 5.22b, respectively.

$$\frac{1}{V}\frac{dN_c}{dt} = \sum_{r\in\mathcal{R}} s_{c,r} r_r, \quad \forall c \in \mathcal{C} \tag{5.22a}$$

$$\frac{dT}{dt} = \frac{-\sum_{r\in\mathcal{R}} r_r \Delta H_r + Q/V}{\rho c_p} \tag{5.22b}$$

where V is the working volume, N_c is the amount of component c, $s_{c,r}$ is the stoichiometric coefficient of product c in reaction r, r_r is the rate of reaction of r, T is the temperature, ΔH_r is the reaction enthalpy, Q is the heat input, ρ is the mixture molar density, and c_p is the specific heat capacity.

Table 5.2: Parameters for Example 1 – single reaction case.

reaction	$s_{A,r}$	$s_{B,r}$	$s_{C,r}$	ΔH_r $[kJ/mol]$	k_r $[10^6/h]$	$E_{A,r}$ $[kJ/mol.K]$
1	-1	1	0	13.8	5.0	50.8
-1	1	-1	0	-13.8	0.1	37.0
2	0	-1	1	-2.0	0.5	46.0

The rate expressions for all three reactions are given by Eq. 5.23.

$$r_r = k_r \exp\left(-\frac{E_{A,r}}{RT}\right) \prod_{c \in rxn_r} \frac{N_c}{V}, \quad \forall r \in \mathcal{R} \tag{5.23}$$

where k_r is the pre-exponential term of the Arrhenius equation, $E_{A,r}$ is the activation energy, R is the ideal gas constant, and rxn_r is the set of components that take place in reaction r. Note that all reactions are assumed to be first order with respect to the reactants.

Lastly, the density and the heat capacity of the mixture are determined by assuming ideal conditions.

$$\begin{aligned} \rho &= \frac{\sum_{c \in \mathcal{C}} N_c}{V} \\ c_p &= \frac{\sum_{c \in \mathcal{C}} c_{p,c} N_c}{\sum_{c \in \mathcal{C}} N_c} \end{aligned} \tag{5.24}$$

The parameters of the reaction conditions are provided in Table 5.2. Notice that the desired reaction is endothermic, therefore heat input to the reactor is mandatory to maintain a certain temperature set-point.

5.3.1.1 Open loop dynamic optimization for optimal trajectories

Let the objective of the batch reaction be to produce a certain amount of product B, while the batch time is minimized. A dynamic optimization problem can be formulated to address such a

makespan minimization problem as presented by Eq. 5.25.

$$
\begin{aligned}
\min_{Q(t)} \quad & t_f \\
s.t. \quad & \frac{1}{t_f}\frac{1}{V}\frac{dN_c}{dt} = \sum_{r\in\mathcal{R}} s_{c,r} r_r, \quad \forall c \in \mathcal{C} \\
& \frac{1}{t_f}\frac{dT}{dt} = \frac{-\sum_{r\in\mathcal{R}} r_r \Delta H_r + Q/V}{\rho c_p} \\
& Eqs.\ 5.23\ and\ 5.24 \\
& N_B(t=1) \geq N_B^{dem} = 0.4\ [kmol] \\
& -12\ [kJ/h] \leq Q(t) \leq 12\ [kJ/h] \\
& N_A(t=0) = 0\ [kmol], \quad N_B(t=0) = 1.0\ [kmol] \\
& N_C(t=0) = 0\ [kmol], \quad T(t=0) = 363\ [K]
\end{aligned}
\tag{5.25}
$$

where N_B^{dem} is the targeted amount of product at the end of the batch. Here, the horizon of the problem is set as $t = [0, 1]$, and the differential equations are scaled by t_f, which denotes the batch time.

The dynamic optimization problem formulated in Eq. 5.25 is solved by orthogonal collocation on finite elements [216]. The orthogonal collocation formulation used in this work is based on Lagrange polynomials and Radau roots. In this case study, we use 24 finite elements (25 mesh points) with 3 collocation points for the differential and algebraic variables. After discretization, the resulting NLP is solved by the IPOPT solver [217]. The optimal open loop profiles are determined as presented in Fig. 5.2, and the minimized makespan t_f is 1.96 hours. The calculated batch time assumes complete degrees of freedom over the manipulated variables. However, practical applications often use closed-loop controllers that manipulate such variables based on process measurements and set points, which results in inherently different process dynamics regardless of the efficacy of the controller. Therefore, the only degrees of freedom in a closed-loop process are the set point trajectories that are transmitted to the controller. The following discussions will focus on (i) developing an MPC scheme for the batch process described by Eqs. 5.22-5.24, (ii) the

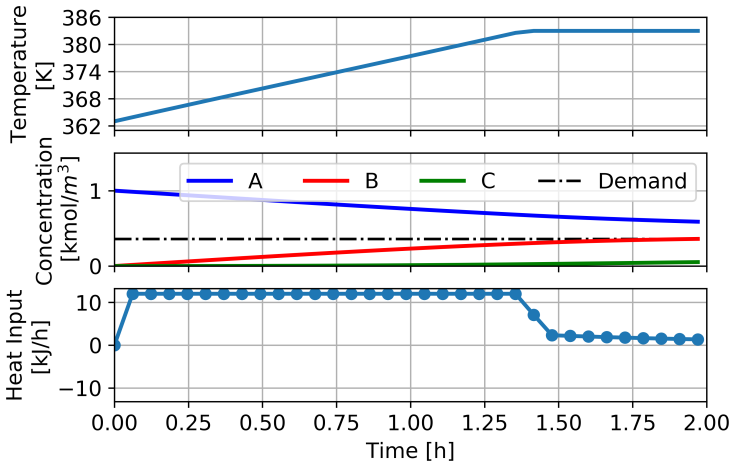


Figure 5.2: Open loop optimal profiles for the temperature, component concentrations, and heat input for Example 1.

effects of the MPC dynamics on the closed-loop realization of the optimal open loop profile, and (iii) accounting for such effects in developing a realizable optimal closed-loop profile.

5.3.1.2 Developing an explicit MPC strategy for Example 1

We follow the procedure described in Section 2.2 to develop an explicit MPC strategy based on the high fidelity model described by Eqs. 5.22-5.24. A multitude of computational experiments are conducted to generate the relevant data for system identification. In each experiment, the input signal is based on a pseudo-random binary sequence (PRBS) and randomized step amplitudes. Using the MATLAB System Identification Toolbox, we develop the following matrices for the approximate model state space model described by Eq. 2.2.

$$A = \begin{bmatrix} 0.9905 & 0.0274 & -0.0299 \\ -0.1174 & 0.9713 & 0.0063 \\ -0.1309 & 0.0618 & 0.9084 \end{bmatrix}$$

$$B = \begin{bmatrix} -0.0860 & -0.1808 & -0.5156 \end{bmatrix}^T$$

$$D = \begin{bmatrix} 0.2735 & 0.2436 & -0.1495 \\ 1.2904 & 0.4882 & -0.4445 \end{bmatrix}$$

Here, the identified model has one input variable, Q, two output variables N_B and T, and three identified states with no significant physical meanings with a discretization step of 0.25 h. The input and outputs are scaled between 0 and 1 to avoid any numerical problems in deriving the explicit MPC solution. Note that C, E, and F from Eq. 2.2 are zero matrices since there are no measured disturbances and zeroth order inputs to the system. The step response of the identified model is provided in the Appendix, Fig. A.5.

The state space model is used in Eq. 2.3 to construct the offline MPC formulation. The bounds on the variables and the tuning of the weight matrices are presented in Table 5.3. Note that the identified states do not have any bounds or weights in the objective function since they have no physical meanings. We also enforce terminal constraints at the end of the output horizon to guarantee the product quality at the process control level by using Eq. 5.26.

$$0.99 y_{k_f}^{sp} \leq y_{k_f} \leq 1.01 y_{k_f}^{sp} \tag{5.26}$$

The constructed mpMPC scheme is rearranged into a generic mpQP problem via the YALMIP toolbox [195] and solved by using the POP toolbox [161] in the MATLAB environment to derive the offline solution in the form of Eq. 2.4. The resulting offline control law has 955 critical regions[2], which requires 955 binary variables for every time step in the horizon to embed in an integrated problem via the standard big-M or convex hull relaxation reformulations. Such a large

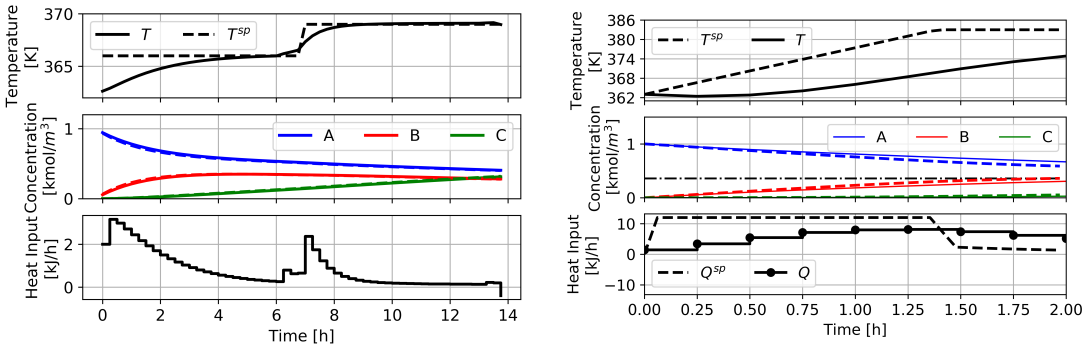
[2]The offline solution can be downloaded from http://paroc.tamu.edu/.

Table 5.3: mpMPC tuning parameters and variable bounds.

parameter	value
OH	$\{1, 2, 3, 4\}$
CH	$\{1, 2, 3\}$
$QR_k, \forall k \in OH$	$\begin{bmatrix} 0 & 0 \\ 0 & 10^4 \end{bmatrix}$
$R1_k, \forall k \in OH$	1.0
$\overline{y}$	$[1.0, 1.0]^T$
$\overline{u}$	1.0
$\overline{\Delta u}$	0.20
$\underline{y}$	$[0, 0]^T$
$\underline{u}$	-1.0
$\underline{\Delta u}$	-0.20

number of binary variables make the integrated problem intractable even to determine a feasible solution. Therefore, we use the base-2 numeral system detailed in Section 5.2.2 to use 10 binary variables for each time step instead.

The developed mpMPC is integrated in the high fidelity model for a closed-loop validation of the developed control law. The closed-loop system is tested rigorously with a set of computational experiments, where the set points are changed arbitrarily, to observe the set point tracking efficacy of the controller. Figure 5.3a presents a sample of a closed-loop simulation, where the temperature set point changes after 7 hours in the operation. The mpMPC scheme achieves satisfactory set point tracking within the range of operation. However, it should be noted that any shift in the operating set point results in a transition period where the states are distant from the desired values, regardless of the effectiveness of the controller. Open loop dynamic optimization approaches neglect these transition periods and assume perfect control over the process. Neglecting the dynamics introduced by the feedback controller may result in a significant mismatch between the desired set point and the realization of the control law. Therefore, we subject the closed-loop system to the open loop optimal set point trajectories presented in Fig. 5.2 to test the compatibility of the controller and operationally relevant conditions. However, the closed-loop simulation fails to run due to infeasible parameter realizations in the mpMPC during the operation. The open loop



(a) Closed-loop validation of the developed mpMPC against the high fidelity model. The temperature set point changes after 7 hours in the operation.

(b) Closed-loop profile of the system subjected to the optimal profile in Fig. 5.2. The terminal constraints of the mpMPC are removed to develop a realizable profile.

Figure 5.3: Closed-loop simulations of the process subjected to (a) arbitrarily changing set points (b) the optimal profile. The dashed lines denote the set points.

optimal profile is unattainable for the controller due to the terminal constraints given by Eq. 5.26. Therefore, we omit these terminal constraints to acquire a feasible closed-loop profile, as presented in Fig. 5.3b. Here, we can observe that the change in the temperature set point is in fact too steep for the controller to track, resulting in infeasible parameter realizations. The open loop optimal trajectory aims to produce the targeted 0.40 kmol/m^3 product B in 1.96 hours, while the achieved yield in closed-loop simulation is 0.349 kmol/m^3, indicating an error of 12.5% mismatch below the desired amount.

With the motivation to bridge the gap between the optimal trajectories and the closed-loop realizations, we integrate the mpMPC dynamics in the dynamic optimization formulation using the base-2 numeral system. The integrated model is first used to determine the maximum possible yield in 2 hours (~1.96 h) using the given process and the developed controller. The resulting MIDO problem is discretized using 8 finite elements (9 mesh points) with 3 collocation points using the Pyomo environment [218, 219, 220]. Note that each finite element has a horizon of 0.25 hours, matching the discretization step of the mpMPC. Discretizing the MIDO yields an MINLP problem, which is solved with GAMS/BARON [221] with a 15 minute limit on the solution time. The time limit is enforced to mimic a real life application, where a decision has to be made periodically.

Accordingly, determining a feasible solution is prioritized over its optimality to guarantee the operability of the process. Figure 5.4a shows the closed-loop dynamic optimization profiles against its implementation on the original high fidelity model. Here, the optimal trajectory and the optimal set points are two distinct entities. While the former is the prediction of the closed-loop profile, the latter indicates the set of operating points that are transmitted to the mpMPC. Notice that the realization of the set points yields a similar profile to that predicted by the dynamic optimization.

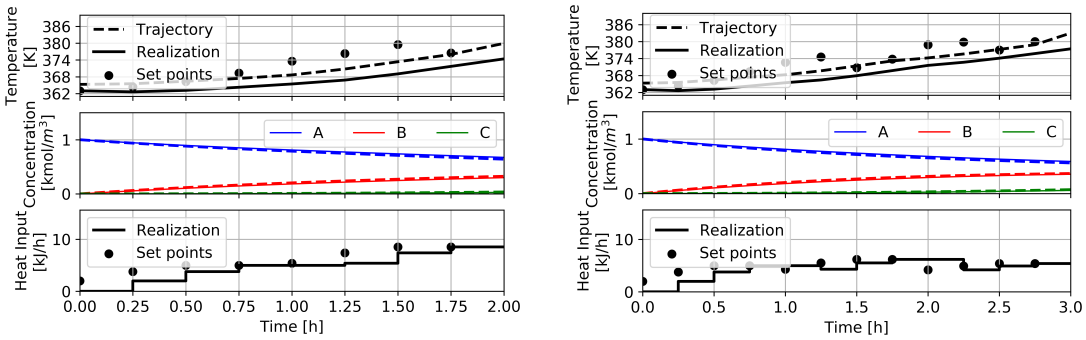
The solution of the MINLP indicates that the maximum possible yield is 0.325 kmol/m^3 at the end of the 2 hour horizon. This result reveals that the original target of 0.40 kmol/m^3 product B in 1.96 hours is in fact infeasible in closed-loop, although an open loop optimal trajectory is attainable. The yield attained at the end of the horizon is 0.314 kmol/m^3 by simulating the closed-loop system against the optimal trajectory, indicating an error of 3.4% mismatch. Note that increasing the number of collocation points per finite element may decrease the error at the expense of increasing the computational complexity.

In Figure 5.4b, the yield of product B is maximized for a horizon of 3 hours using the same explicit control strategy. In this problem, 12 finite elements (13 mesh points) are used to maintain the horizon of each individual element to 0.25 hours to match with the time steps of the mpMPC. The dynamic optimization formulation predicts a yield of 0.367 kmol/m^3, while the closed-loop simulation tracking the optimal profile produces 0.363 kmol/m^3 product B with an error of 1.1%.

The proposed integration methodology consistently bridges the gap between the optimal profiles that are used as set points by the controllers and the actual output of the closed-loop process, as demonstrated in Fig. 5.4. The explicit solution of the MPC scheme facilitates its exact implementation into a dynamic optimization formulation. Moreover, the base-2 numeral system is used as a basis for the relaxation of the piecewise affine critical regions, which rendered the problem computationally tractable by exponentially reducing the required number of binary variables.

5.3.2 Three reactions in two reactors

In this case study, we consider a system of three sets of reactions taking place in two reactors, where both reactors are capable of processing the available tasks. The stoichiometry of the



(a) 2 hours.

(b) 3 hours.

Figure 5.4: Closed-loop dynamic optimization to maximize the yield of B in given time and the validation of the optimal profile against the high fidelity model. "Set points" denote the targets determined by the MIDO and used by the mpMPC in closed-loop, "Trajectory" represents the closed-loop profile that is predicted by the MIDO, "Realization" denotes the actual closed-loop profile observed in the simulation.

reactions is presented as follows.

$$\text{Reaction 1: } A \underset{k_{-1}}{\overset{k_1}{\rightleftharpoons}} B \xrightarrow{k_2} C$$

$$\text{Reaction 2: } A \xrightarrow{k_3} D$$

$$\text{Reaction 3: } B + D \xrightarrow{k_4} E$$

where the first reaction set has the dynamics from the previous case study. In this example, the valuable products are B, D, and E. Therefore, the operator has the degree of freedom to select the most convenient task that delivers the requirements of the desired objective at a given time. We employ the SEN framework, discussed in Section 5.2.1, to determine the process schedule over a given horizon. The SEN representation of the process is shown in Fig. 5.5. The mathematical model presented in Eqs. 5.22-5.24 are used to simulate the dynamic behavior of the system, and the parameters of Reaction 2 and 3 are provided in Table 5.4.

The objective of this problem is to maximize the profit, which accounts for the revenue from selling the products, the operating cost for the heat supply and raw material purchases, and the investment cost due to the sizing of the reactors. The design decisions are particularly included in

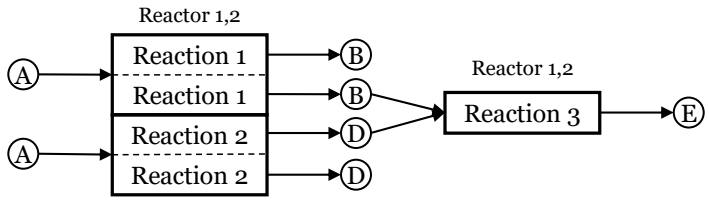


Figure 5.5: State equipment network representation of the process in Example 2.

Table 5.4: Parameters for Example 2 – three reactions and two reactors case.

	ΔH_r $[kJ/mol]$	k_r	$E_{A,r}$ $[kJ/mol.K]$
Reaction 2	9.8	$5.0 \times 10^3\ h^{-1}$	90
Reaction 3	-10	$0.5\ m^3 kmol^{-1} h^{-1}$	40.7

this case study, as they directly conflict with the operating decisions in terms of the optimality and feasibility of the problem. Note that the interactions between the design and operating decisions do not conflict in other objectives such as maximizing the yield or minimizing the makespan. One can simply maximize the reactor volumes to increase these performance metrics, which will yield unrealistic design solutions.

We follow the same procedure to develop the explicit MPC strategies for Reactions 2 and 3. The approximate state space models for these systems and their step responses are provided in Appendix A.5. The tuning parameters of both control strategies are presented in Table 5.5. The variable bounds, and the output and control horizons are omitted here since they are identical to the values in Table 5.3. The explicit MPC solutions have 647 and 977 critical regions for Reactions 2 and 3, respectively[3].

In this example, the schedule is designed over a horizon of 8 hours. We assume that the processing time for the separation of the product of interest from the unreacted raw materials and by-products is negligible. The integrated MIDO problem is reformulated as an MINLP by ortogonal collocation on finite elements with 24 finite elements and 3 collocation points over each finite

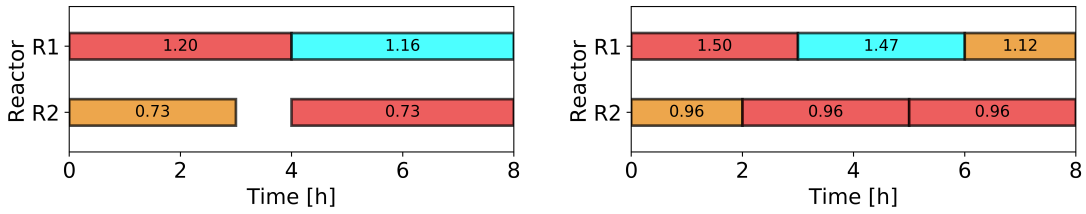
[3]The explicit MPC solutions can be downloaded from http://paroc.tamu.edu/.

Table 5.5: mpMPC tuning parameters for Reactions 2 and 3 in Example 2.

parameter	Reaction 2	Reaction 3
$QR_k, \forall k \in OH$	$\begin{bmatrix} 0 & 0 \\ 0 & 10^3 \end{bmatrix}$	$\begin{bmatrix} 0 & 0 \\ 0 & 4 \times 10^3 \end{bmatrix}$
$R1_k, \forall k \in OH$	1.0	1.0

element. The reformulated MINLP is solved with GAMS/BARON [221] with a 15 minute limit on the solution time. The resulting process schedule is demonstrated on a Gantt chart in Fig. 5.6a. By the end of the scheduling horizon, the targeted inventory is 0.28 kmol for B and 0.39 kmol for E, while no excess D is produced. For reference, the same problem is solved without accounting for the dynamics introduced by the MPC, which is presented in Fig. 5.6b. Here, the scheduler aims to produce 0.76 kmol B, 1.02 kmol D, and 0.49 kmol E by the end of the horizon. Notice that the targeted inventory levels are in fact lower when the MPC dynamics are included in the integrated problem. Acquiring "worse" solutions with smaller profit margins with the proposed approach is an expected outcome since the problem without the MPC dynamics is an underestimator[4] of the completely integrated problem. The most imperative contribution of the proposed approach is to provide certificates of operability for the calculated optimal trajectory under the simultaneously determined process design. The benefit of having such certificates can be observed in Fig. 5.7, where the optimal trajectories are determined based on closed-loop and open loop dynamic optimization formulations for three distinct tasks. The proposed approach bridges the gap between the optimal trajectory and its realization in closed-loop by distinguishing the desired path from the set points that need to be passed on to the controller. Following these trajectories, the proposed scheduling and control scheme achieves the targeted inventory levels by producing 0.28 kmol B and 0.38 kmol E at the end of the horizon. However in the reference case, the controllers fail quite often due to infeasible parameter realizations. The steep changes in set points impose unrealistic trajectories for the controller, which cannot satisfy the terminal constraints. Therefore, the closed-loop realizations in Figures 5.7b, 5.7d, and 5.7f are simulated without enforcing the terminal constraints in

[4]Underestimator is used in the direction of a conventional minimization problem.



(a) Process schedule with the MPC dynamics included. The reactor sizes are $V_{R1} = 1.20m^3$ and $V_{R2} = 0.73m^3$, respectively.

(b) Process schedule without the MPC dynamics. The reactor sizes are $V_{R1} = 1.50m^3$ and $V_{R2} = 0.96m^3$, respectively.

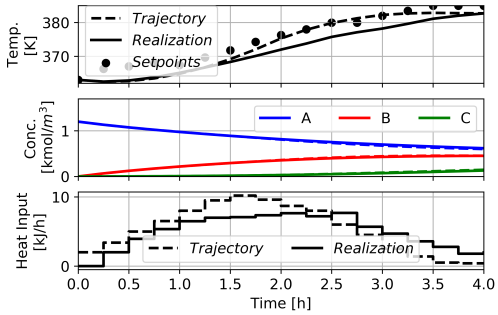
Figure 5.6: Process schedule with and without the MPC dynamics. The colors red, orange, and cyan represent the production of B, D, and E, respectively.

the MPC. Due to the mismatch between the set points and the closed-loop realization, the targeted production cannot be achieved in the dedicated time interval. In other words, the schedule cannot be satisfied due to the delay in delivering the intended amounts.

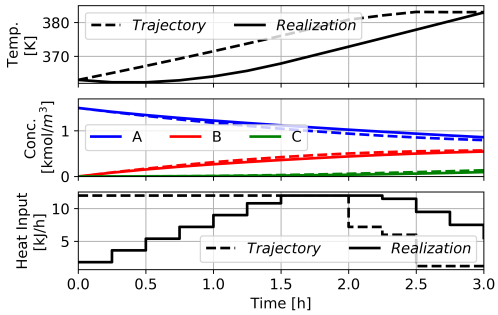
5.4 Conclusion

In this work, we presented a comprehensive modeling methodology to integrate the process design, scheduling, and advanced control decisions in a single optimization formulation for a batch process, while accounting for the closed-loop operability. We introduced an exact formulation technique to integrate the MPC dynamics into a mixed integer dynamic optimization problem by multi-parametric programming. The piecewise affine expression that represents the offline look-up table for the optimal control law is embedded via the base-2 numeral system, which exponentially reduced the required number of binary variables for the formulation. The scheduling problem was formulated with the SEN framework due to its suitability for the integration with dynamic optimization problems with dynamic optimization problems through logical disjunctions.

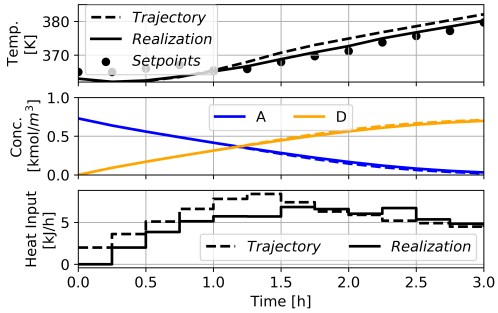
The main goal of this study is to account for the dynamics introduced by an MPC scheme acting on a batch process. Open loop dynamic optimization often neglect such shifts in dynamic behavior. However, it was observed that the MPC scheme may be unable to track the optimal open loop trajectory due to its implicit effects on the process dynamics. Failing to deliver the targeted production rates at the lower level decision layers caused the failure of several scheduling decisions



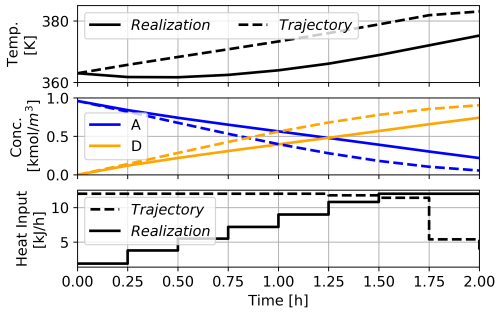
(a) Production of B based on closed-loop optimization.



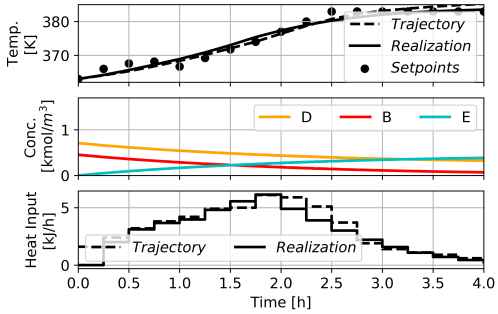
(b) Production of B based on open loop optimization.



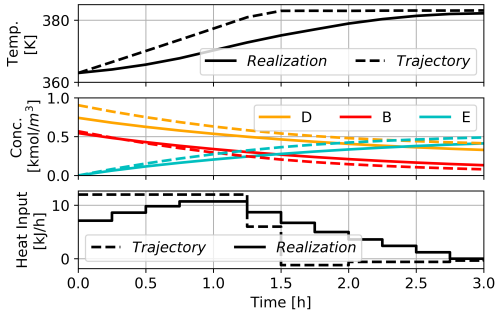
(c) Production of D based on closed-loop optimization.



(d) Production of D based on open loop optimization.



(e) Production of E based on closed-loop optimization.



(f) Production of E based on open loop optimization.

Figure 5.7: Closed-loop validation of the optimal input-output trajectories for three different tasks based on the closed-loop and open loop dynamic optimization formulations. Figures on the left hand side (5.7a, 5.7c, and 5.7e) show the optimal closed-loop trajectories, set-points that are passed on to the MPC, and the closed-loop realizations. Figures on the right hand side (5.7b, 5.7d, and 5.7f) only show the optimal trajectories and their closed-loop realizations since the trajectories are used as the set points.

such as equipment assignment and task start/end times in closed-loop application. Integrating the feedback control law in the dynamic optimization allowed for deriving more attainable closed-loop trajectories and thus, realizable processing times for the scheduler.

The proposed methodology was tested with different objectives, namely makespan minimization, yield maximization, and profit maximization. We showcased two case studies, the latter of which comprised all the decision layers in the scope of this work, i.e. process design, scheduling, and control, with the objective to maximize the profit. Although the closed-loop optimization predicts lower revenues and higher costs than the open loop trajectories, the optimized profiles were realized with significantly higher accuracy. Therefore, accounting for the MPC dynamics is paramount to provide certificates of operability.

Our future efforts will focus on addressing industrially relevant batch processes, which are much larger in problem size. Although the proposed methodology allows for finding a feasible solution, its optimality still needs significant improvement to approach the global minimum. Our future efforts will include two main avenues that can also be used in tandem to improve the solution quality. First, the piecewise affine control law is quadratic (hence convex) in the objective space. A tailored branch and bound algorithm can be developed to fathom the infeasible or suboptimal control laws by benefiting from the structure of the multi-parametric solution space. Second, the open loop optimal trajectory is an underestimator of the integrated problem as discussed in Section 5.3. This solution can be used in the tailored algorithm to achieve a better initial starting point. An improved solution strategy is essential to solve larger scale and industrially relevant problems.

Further improvement can be achieved with a more accurate representation for the nonlinear effects of the design variables in the process control level. In our earlier works [105, 222, 88], we accounted for the design variables as added disturbances in the MPC scheme. However, highly nonlinear process design variables require piecewise affine models that render the mpMPC problem into an mpMIQP, which significantly increases the size of the offline look-up table. Our current efforts focus on treating the design variables as left hand side parameters instead of added disturbances to alleviate the computational burden while handling highly nonlinear process designs.

6. A NOVEL ALGORITHM FOR MULTIPARAMETRIC PROGRAMMING[1]

6.1 Introduction

Multiparametric programming is an established tool to solve optimization problems in the presence of uncertain parameters [223]. The advantages of multiparametric programming lie in the offline map of optimal solutions that (i) provides valuable insight on the behavior of the optimal decision under a range of parameters prior to their realization, (ii) the burden of solving an optimization problem is removed and replaced with evaluating an explicit function after the realization of the parameters [224], and (iii) allows for an exact formulation to embed the optimal solutions in the context of simulation and optimization [8]. The range of applications for multiparametric programming spans multiparametric/explicit Model Predictive Control (mpMPC) [85, 225, 226, 227, 228, 229], process scheduling [230, 187], integration of multi-scale decisions [177, 88, 106, 105, 222, 8, 231], bi-level programming [232, 233], and parameter estimation [234].

Complete theories and solution strategies were proposed in the literature for multiparametric linear programming (mpLP), quadratic programming (mpQP) [235, 85, 236, 237, 238, 239, 240], and mixed-integer linear programming (mpMILP) [241]. A key difference in existing approaches is their procedure to explore the parameter space to completion. Algorithms proposed by Bemporad *et al.* (2002) [85], Tøndel *et al.* (2003) [236], and Spjøtvold *et al.* (2006) [237] rely on geometrical strategies, where the parametric solution is determined by direct exploration of the parameter space. Strategies for multiparametric programming proposed by Gupta *et al.* (2011) [238] and Ahmadi-Moshkenani *et al.* (2018) [239] develop the parametric solution by enumerating possible active set combinations with a branch and bound style approach. These active set strategies are inherently different from geometrical approaches because they do not rely on the parameter space to identify the optimal explicit expressions that are defined over the parameter space. Algorithms that incorporate both geometric and active set strategies, by Gal *et al.* (1972) and Oberdieck

[1]Portions of this chapter have been submitted for publication as Burnak, B., Katz, J., Pistikopoulos, E.N., A novel geometrical algorithm for multiparametric programming, *in review*.

et al. (2017) [235, 240], rely on representing the parametric solution as a connected graph where each node represents an optimal active set combination.

Although these approaches theoretically guarantee developing the complete solution over the parameter space, practical implementation becomes more challenging as the number of optimization variables, constraints, and parameters grow because of the potential exponential increase in optimal active set combinations. Managing the memory requirements of an exponential solution space has been approached by Drgoňa *et al.* (2017) via the so-called regionless explicit MPC [242]. The regionless explicit MPC strategy saves memory by maintaining factored matrices and active set combinations, instead of the optimal expressions defined over the parameter space. However, with a solution space that grows combinatorially with the problem size, developing the full parametric solution becomes impractical, and using the complete explicit solution in offline applications becomes intractable. For instance, in multi-level optimization formulations, the solution space of the follower (lower level) problems increase rapidly in the number of variables and constraints, necessitating a strategy to account for the potential explosion of optimal active set combinations that define the multiparametric solution. Current theory and strategies in the open literature do not attempt to address this potential explosion, hence the use of explicit solutions in large scale offline applications is rather limited. Therefore, the exploration of a meaningful partial solution to these large scale problems is necessary. In other words, the question that must be addressed is "What is a good criterion that provides meaningful insight to the multiparametric solution, and how can an efficient strategy be implemented to exploit this criterion?".

In this chapter, we propose a novel parameter space exploration algorithm for mpLP, mpQP, and mpMILP based on recursive construction of nonincreasing simplices via Delaunay triangulation. The proposed algorithm prioritizes the volumetrically larger partitions of the solution space, whereas the existing multiparametric programming solvers place no priority for the size of the partitions identified. Identifying the larger partitions is particularly important in large scale multi-level optimization problems, where even finding a feasible solution can be challenging. In these problems, having identified the larger partitions of the follower (lower level) optimization prob-

lem facilitates finding a feasible overestimator of the global minimum, while keeping the problem tractable. The proposed algorithm returns a larger portion of the parameter space compared to the existing state-of-the-art multiparametric solvers upon early termination, which is a promising step towards using explicit optimal solutions in large scale offline applications.

The remainder of the chapter is organized as follows. A brief overview of multiparametric programming is presented in Section 6.2. In Section 6.3, the proposed algorithm is described. The performance of the proposed approach is evaluated by numerical examples and compared against state-of-the-art solvers in Section 6.4. Lastly, a summary of the paper and directions for future work are presented in Section 6.5.

6.2 Preliminaries for multiparametric programming

We consider standard mpLP and mpQP problems, described in the following general form given in Problem P1. Note that the discussions will be extended to mixed-integer problems in Section 6.3.5.

$$\begin{aligned} z^*(\theta) = \quad & \min_{x} \quad (Qx + H\theta + c)^T x \\ & s.t. \quad Ax \leq b + F\theta, \; \theta \in \Theta \end{aligned} \tag{P1}$$

where $x \in \mathbb{R}^n$ is the vector of optimization variables, $\theta \in \mathbb{R}^q$ is the vector of parameters defined in a convex polytope $\Theta \subset \mathbb{R}^q$, $z^*(\theta) \in \mathbb{R}$ is the optimal objective value as a function of the parameters θ, and $Q \succ 0 \in \mathbb{R}^{n \times n}$, $H \in \mathbb{R}^{n \times q}$, $c \in \mathbb{R}^n$, $A \in \mathbb{R}^{m \times n}$, $b \in \mathbb{R}^m$, $F \in \mathbb{R}^{m \times q}$. Note that Q is defined for mpQP problems only. Also, let $f(x, \theta)$ denote the objective function, $\mathcal{N} \triangleq \{1, 2, ..., n\}$ denote the set of optimization variables, and $\mathcal{M} \triangleq \{1, 2, ..., m\}$ denote the set of indices of all constraints in Problem P1 in the following discussions.

Definition 1. (Linear Independence Constraint Qualification (LICQ) [237]). Let $\mathcal{A}$ indicate the index of active constraints at any parameter realization $\bar{\theta}$. LICQ holds if the set of active constraint gradients is linearly independent, i.e. $A_{\mathcal{A}}$ has full row rank.

Definition 2. (Strict Complementarity Slackness (SCS) [237]). Let $x^*(\bar{\theta})$ be the optimal solution, and $\lambda^*(\bar{\theta})$ be the set of Lagrange multipliers for a parameter realization $\bar{\theta}$. SCS holds if either the i^{th} constraint in Problem P1 is active ($A_i x^*(\bar{\theta}) = b_i + F_i\bar{\theta}$) or the corresponding Lagrange multiplier is zero ($\lambda_i^*(\bar{\theta}) = 0$) for each $i \in \mathcal{M}$.

Theorem 1. (Basic Sensitivity Theorem [243]). *Let $x^*(\bar{\theta})$ be the optimal solution and $\lambda^*(\bar{\theta})$ be the set of Lagrange multipliers at any parameter realization $\bar{\theta}$. Also assume that LICQ and SCS hold. Then, there exists a unique, once differentiable function $[x^{*T}(\theta), \lambda^{*T}(\theta)]^T$ satisfying the Karush-Kuhn-Tucker (KKT) optimality conditions in the neighborhood of $\bar{\theta}$, and*

$$\begin{bmatrix} x^*(\theta) \\ \lambda^*(\theta) \end{bmatrix} = -M^{-1}N(\theta - \bar{\theta}) + \begin{bmatrix} x^*(\bar{\theta}) \\ \lambda^*(\bar{\theta}) \end{bmatrix} \tag{1}$$

where

$$M = \begin{bmatrix} \nabla^2_{xx}\mathcal{L} & \nabla_x g_1 & \cdots & \nabla_x g_m \\ \lambda_1 \nabla_x^T g_1 & g_1 & & \\ \vdots & & \ddots & \\ \lambda_m \nabla_x^T g_m & & & g_m \end{bmatrix}$$

$$N = [\nabla^2_{\theta,x}\mathcal{L}, \lambda_1 \nabla_\theta^T g_1, \lambda_m \nabla_\theta^T g_m]^T$$

$$\mathcal{L}(x, \lambda, \theta) = f(x, \theta) + \lambda^T g(x, \theta)$$

$$g(x, \theta) = Ax - b - F\theta$$

Remark 1. If the quadratic term Q is not defined, Problem P1 describes an mpLP, where Theorem 1 also holds without loss of generality.

Definition 3. (Piecewise affine [85]). A function $x(\theta) : \Theta \subset \mathbb{R}^q \to \mathbb{R}^n$ is piecewise affine if it is possible to partition Θ into polytopic regions, such that

$$x(\theta) = K_j\theta + r_j, \forall \theta \in \Omega_j, j \in J \tag{2}$$

where Ω_j is defined as the j^{th} polytopic region, and J is the index set. Note that piecewise quadratic is defined analogously.

Theorem 2. (Properties of mpQP solution [85, 244]). *Consider the mpQP problem presented in Problem P1, where $Q \succ 0$. Then, the set of feasible parameters $\Theta_f \subseteq \Theta$ is convex, the optimizer $x^*(\theta)$ is continuous and piecewise affine, and the optimal objective function $z^*(\theta)$ is continuous and piecewise quadratic.*

Remark 2. Without loss of generality, Theorem 2 holds for mpLP solutions except the optimal objective function $z^*(\theta)$ is piecewise affine [245, 85].

Definition 4. (Critical region). A polytopic region Ω_j is a critical region, denoted by CR_j, if Eq. 2 describes the optimal solution to Problem P1.

Lemma 1. *Each critical region CR_j is uniquely defined by the optimal active set associated with it [238].*

6.3 A Novel Geometric Exploration Strategy

We propose a systematic sampling strategy via Delaunay triangulation for the parameter search space that prioritizes the volumetrically large critical regions. We begin the discussion by defining "candidate subset" and "candidate simplex", which are the building blocks of the proposed algorithm.

Definition 5. (Candidate subset and candidate simplex). Any full-dimensional polytope that is a subset of $\Theta \subset \mathbb{R}^q$ is a *candidate subset*, $\Theta_c \subseteq \Theta$. If the subset has $q+1$ vertices, the candidate subset is a *candidate simplex*.

The proposed strategy relies on (i) constructing candidate simplices in a non-increasing sequence, (ii) identifying the candidate simplices that are subsets of the optimal partitions in the parameter space, and (iii) selecting a new sampling point if the candidate simplex is not a subset of an optimal partition. The critical region around the sampled parameter realization is constructed based on the Basic Sensitivity Theorem [243]. The procedure to explore the parameter space and

developing the optimal partitions is summarized in Algorithm 1. We describe the detailed steps of the exploration algorithm in Section 6.3.1.

Algorithm 1 *Parameter space exploration procedure*

1: Get Problem P1 and the parameter space Θ. Let $\Theta_c^h \leftarrow \Theta$.
2: Solve Problem P1 at all the vertices of Θ_c^h.
3: If the set of strongly active constraints, $\overrightarrow{\mathcal{A}}$, is identical for all vertices, eliminate Θ_c^h from the parameter search space. Else, proceed to Step 5.
4: Check $\overrightarrow{\mathcal{A}}$ for dual degeneracy. If it is non-degenerate, construct CR_j based on the Basic Sensitivity Theorem (see Theorem 1) (Also see Section 6.3.4 for degenerate cases), and proceed to Step 7.
5: Let p_c be the center of mass of $\overrightarrow{\theta}^h$, $\langle \overrightarrow{\theta}^h \rangle$.
6: Determine the child simplices of the point set $\overrightarrow{\theta}^h \cup p_c$ via Delaunay triangulation (see [246] for the details on Delaunay Triangulation). Include the child simplices in the set of candidate convex subsets.
7: Go back to Step 2. Repeat for all parent candidate convex subsets.
8: Increment h.

6.3.1 Parameter space exploration

Assume the solutions to Problem P1 are feasible at all vertex points of Θ. Note that this assumption will be relaxed in Section 6.3.3, where an initialization strategy is presented.

The exploration procedure is initialized ($h = 0$) by defining a candidate convex subset, Θ_c^h equal to the parameter space Θ. The initial candidate convex subset dictates the boundaries of the search space throughout the rest of the procedure, because the algorithm explores the space by creating new candidate simplices in a non-increasing sequence in the subsequent iterations. Therefore, it is guaranteed that the algorithm never explores outside the parameter bounds.

Problem P1 is solved at the vertex points of Θ_c^h to find the corresponding active sets, $\overrightarrow{\mathcal{A}}$. Lemma 1 suggests that if all active set combinations in $\overrightarrow{\mathcal{A}}$ are identical at the set of vertex points $\overrightarrow{\theta}^h$, then $\overrightarrow{\mathcal{A}}$ uniquely defines the critical region bounded by Θ_c^h. Therefore, knowing that the parameter space is explored to completion (i.e. $\Theta \setminus \Theta_c^h = \varnothing$), the exploration algorithm is terminated by eliminating Θ_c^h from the parameter search space, since it comprises only one optimal active set

combination. Note that this is a trivial case where Θ is feasible for all parameter realizations and has one critical region.

In the case where $\overrightarrow{\mathcal{A}}$ at $\overrightarrow{\theta}^h$ are not all identical, there exist at least two critical regions within Θ_c^h by Lemma 1. Therefore, we generate new non-increasing *child* candidate simplices for the next iteration (Θ_c^{h+1}) to explore the parameter space in higher resolution. Although it is possible to generate any finite number of non-overlapping candidate simplices [247], we propose a systematic and efficient procedure to construct the child subsets iteratively. In the proposed algorithm, these child subsets are generated such that (i) they are non-overlapping simplices, and (ii) each child subset has q vertices that belong to the point set $\overrightarrow{\theta}^h$ and share one vertex at an arbitrary point, $p_c \in \Theta_c^h$. An effective methodology to construct such subsets is to utilize computational geometry tools such as triangulation algorithms. In this study, we employ Delaunay triangulation to generate child subsets from the parent subsets (the interested reader is referred to de Berg *et al.* [246] for details on Delaunay triangulation). Although this step can be replaced by any other triangulation algorithm, Delaunay triangulation provides two main benefits. First, due to the empty circle property, it yields well-distributed simplices compared to other algorithms [246], which promotes sparse sampling in the parameter space and thus targets volumetrically larger critical regions. Second, Delaunay triangulation is a well-established technique in the field of computational geometry, and its software implementation is readily available in most of the widely used programming languages.

The procedure to generate child subsets Θ_c^{h+1} from a given parent subset Θ_c^h is depicted in Figure 6.1, where the center of mass of the vertex points is assigned as $p_c = \langle \overrightarrow{\theta}^h \rangle$, where $\langle \cdot \rangle$ represents the center of mass of a point set. The triangulation step is executed for each parent subset of which the active sets, $\overrightarrow{\mathcal{A}}$, are different at the vertex points. On the other hand, if the subset $\overrightarrow{\mathcal{A}}$ is identical at the vertex points, we know that the simplex is a subset of a critical region (i.e. $\Theta_c^h \subseteq CR_j$) by Lemma 1. Therefore, Θ_c^h can be eliminated from the parameter search space since there is no need for further exploration.

The point set $\overrightarrow{\theta}^h$ is checked for dual degeneracy based on its corresponding active sets, $\overrightarrow{\mathcal{A}}$. Handling dual degeneracy is omitted in this section to focus on exploration of the parameter space,

and will be discussed in Section 6.3.4. For the non-degenerate case, the unique combinations of $\overrightarrow{\mathcal{A}}$ and the corresponding parameter realizations $\theta \in \overrightarrow{\theta}^h$ are used to construct the critical regions by the Basic Sensitivity Theorem. We derive the parametric expressions for the optimal solution $x^*(\theta)$ and optimal Lagrange multipliers $\lambda^*(\theta)$ for all $\theta \in \overrightarrow{\theta}^h$ by Eq. 1. The bounds of the critical regions and optimal objective function $z^*(\theta)$ are determined by direct substitution of $x^*(\theta)$ and $\lambda^*(\theta)$ into Problem P1 and $\lambda^*(\theta) \geq 0$.

The points sampled from the parent subset Θ_c^h in iteration h comprise the point set $\overrightarrow{\theta}^{h+1} = \overrightarrow{\theta}^h \cup p_c$ for the next iteration. The generated child subsets, Θ_c^{h+1}, are subjected to the same exploration procedure, until all candidate simplices are eliminated from the parameter search space. Therefore, the presented parameter space exploration algorithm can be summarized as follows.

i. Solve an optimization problem at the vertex points of each candidate subset to determine the minimum number of unique active sets (i.e. critical regions) in the corresponding subsets.

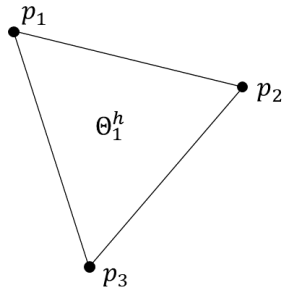
ii. Develop the critical regions around the vertex points by Eq. 1.

iii. If there exists one unique active set in a candidate subset, eliminate the subset from the parameter search space. Else, generate child subsets by Delaunay triangulation.

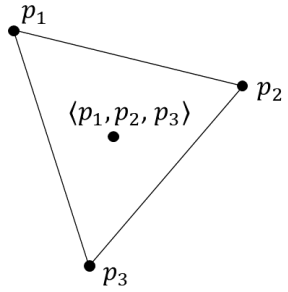
iv. Repeat until all candidate subsets are eliminated from the parameter search space.

Remark 3. One major iteration consists of two main loops to (i) solve the optimization problem at the sampled points, $\overrightarrow{\theta}^h$, and (ii) eliminate the fully explored subsets from the parameter search space and triangulate into finer simplices if necessary. Note that the cycles in these loops are completely independent, i.e. they can be evaluated without requiring the output of another cycle. Therefore, both loops can be executed in parallel if multi-core processors are available.

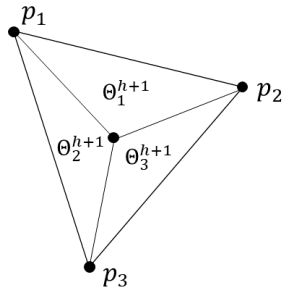
Remark 4. The active set $\overrightarrow{\mathcal{A}}$ is determined by checking for the positive Lagrange multipliers ($\lambda_i > 0$), rather than the classical definition ($g_i = 0$) to exclude the weakly active constraints from the active set. Distinguishing the weakly and the strongly active constraints alleviates the dual degeneracy problem during the exploration step.



(a) Parent candidate simplex Θ_1^h, with vertices $\{p_1, p_2, p_3\}$.



(b) Center of triangulation, $p_c = \langle p_1, p_2, p_3 \rangle$, is defined at the center of mass of the vertex points.



(c) Delaunay triangulation with the new point included in the point set. All Θ_c^{h+1} are siblings to each other, and children of Θ_1^h.

Figure 6.1: The proposed procedure to generate child candidate simplices from a given parent subset by using Delaunay triangulation.

Remark 5. The point set $\overrightarrow{\theta}^h$ may include parameter realizations sampled from an already explored critical region. In that case, solving the optimization problem at these corresponding points is redundant, since we know that the solution will not reveal any new active set combinations by Lemma 1. Therefore, we can benefit from the previously explored critical regions to decide if the solution of the optimization problem at the parameter realization is required.

The presented parameter space exploration strategy provides a structured methodology to sample the solution space. A key benefit of the sampling strategy employed is the ability to prioritize identifying volumetrically larger critical regions. Large critical regions are prioritized because the likelihood a vertex associated with a triangulated child simplex exists within a larger critical region is proportional to its volume. In other words, larger critical regions are likely to be identified compared to smaller critical regions due to the proposed sampling method.

6.3.2 Illustrative example

A demonstration of Algorithm 1 is provided via an mpQP example with two parameters for visualization, 10 optimization variables, and 15 constraints[2]. Here, we also introduce the concept of accumulated volume, which will be discussed further in Section 6.4.2. The problem structure is based on Problem P1, and the defining matrices are provided in the Appendix. The steps below are illustrated in Figure 6.2.

Step 1 Given the upper and lower bounds of the hypercube defined by the polytope Θ, the initial vertices are located, seen by Figure 6.2a, $\overrightarrow{\theta}^{h=0} = \{(10, 10), (10, -10,), (-10, -10), (-10, 10)\}$. For the presented problem, the total volume of the parametric solution is 400 magnitude units (m.u.).

Step 2 An optimization problem is solved at each vertex point defined in Step 1. The solution to these optimization problems provides the optimal active set combination for each vertex. The active set for the vertices are defined as $\overrightarrow{\mathcal{A}} = \{\{12\}, \{7, 13\}, \{7, 14\}, \{3, 14\}\}$. These active sets are not all identical, and by Lemma 1, multiple critical regions must exist. The critical regions

[2]The example problem and its exact solution can be downloaded at http://paroc.tamu.edu/Examples/.

developed in this step occupy an accumulated volume of 104.9 m.u. These critical regions account for 26.2% of the volume of the total solution.

Step 3 The polytope defined by Θ needs further exploration because of the active set discrepancies found in Step 2. First, the center of mass of the vertices is determined, $\langle \overrightarrow{\theta}^{h=0} \rangle = (0, 0)$, and added to the vertex list $p_c = \overrightarrow{\theta}^{h=1} = \langle \overrightarrow{\theta}^{h=0} \rangle$. The initial vertices together with p_c are used to perform Delaunay triangulation, Figure 6.2c, which provides the union of child simplices to be explored, $\Theta_c^{h=1}$.

Step 4 Each child simplex is treated as a convex polytope, similar to the initial set of vertices defined in Step 1. An optimization problem is solved for each newly defined vertex. In this case, the only new vertex added was $\langle \overrightarrow{\theta}^{h=0} \rangle = \{(0, 0)\}$, and therefore only a single optimization problem is solved in this step. The optimal active set combination is $\{\varnothing\}$. The critical region identified has a volume of 97.9 m.u., and the accumulated volume is 202.8 m.u. The accumulated volume accounts for 50.7% of the total volume of the multiparametric solution.

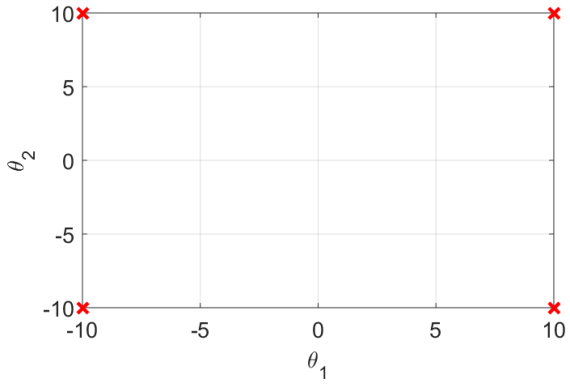
Step 5 Each generated triangle determined in Step 3 is then analyzed. For a given generated triangle, $\Theta_c^{h=1}$, if all of the active set combinations associated with its vertex list are identical, the simplex $\Theta_c^{h=1}$ is eliminated from the parameter search space, otherwise further exploration is needed. For instance, $\Theta_1^{h=1}$ has an active set list of $\overrightarrow{\mathcal{A}} = \{\{\varnothing\}, \{7, 13\}, \{7, 14\}\}$, and thus further exploration is needed. Each generated triangle requires further exploration, and therefore a new set of points are defined $\overrightarrow{\theta}^{h=2} = \{(0, -6.67), (-6.67, 0), (0, 6.67), (6.67, 0)\}$

Step 6 Repeat Steps 2-5 until the termination criterion is met.

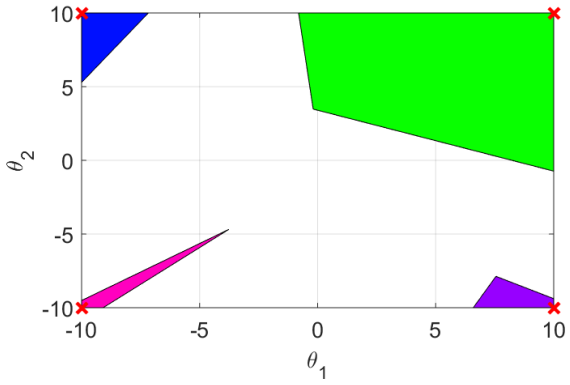
6.3.3 Initialization

In Section 6.3.1, we assumed that the solutions to Problem P1 are feasible at all vertex points of Θ. However, this is rarely the case, and we need an effective initialization procedure when this assumption does not hold.

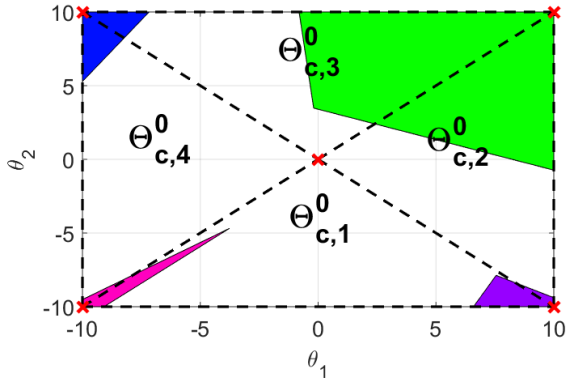
Algorithm 1 can be initialized by labeling the infeasible vertex points of Θ as *infeasible* to distinguish from the feasible $\overrightarrow{\mathcal{A}}$. However, if there exists at least one infeasible parameter realiza-



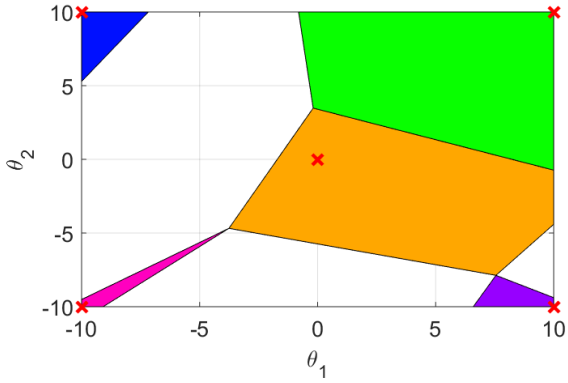
(a) Step 1: Locate initial vertices.



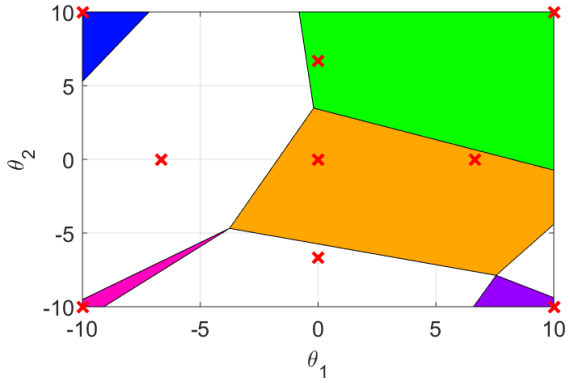
(b) Step 2: Identify active sets.



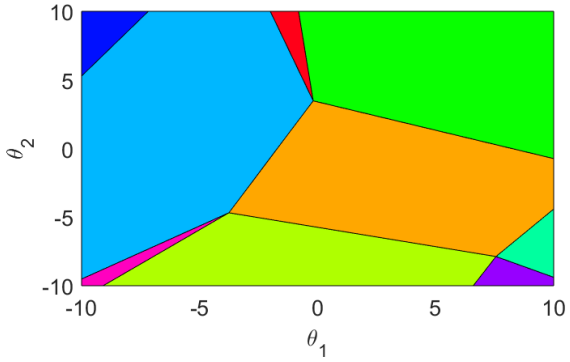
(c) Step 3: Delaunay triangulation.



(d) Step 4: Identify active sets.



(e) Step 5: Further exploration.



(f) Complete map of solutions.

Figure 6.2: Illustration of the proposed algorithm.

tion in the points set $\overrightarrow{\theta}^h$ for $h = 0$, the proposed algorithm cannot guarantee exploration of the full solution. The complement of the feasible parameter space is nonconvex in the general case. Therefore, we cannot effectively use Algorithm 1 to eliminate the infeasible parameter space.

Hence, we propose a slight modification on the elimination criteria to maintain an exact algorithm, as outlined in Algorithm 2, that guarantees the acquisition of the full solution when there exists at least one infeasible vertex in Θ.

For a given candidate simplex Θ_c^h, we compare the active set combination $\overrightarrow{\mathcal{A}}$ as discussed in Algorithm 1. If all the vertex points are labeled as *infeasible*, we need to check if there exists a feasible solution in the parameter space bounded by Θ_c^h. This would be a trivial exercise if we had the closed half-space representation of Θ_c^h, however we only collect the vertex points of the simplices during the exploration of the parameter space. Therefore, we define a point p, which is a convex combination of the vertex points of the given candidate simplex by Theorem 3.

Algorithm 2 *Modified parameter space elimination procedure*

1: Get Problem P1, active set $\overrightarrow{\mathcal{A}}$, and candidate convex subset Θ_c^h.
2: If $\overrightarrow{\mathcal{A}}$ is identical at all the vertices of Θ_c^h and a feasible combination, eliminate Θ_c^h from the parameter search space and terminate Algorithm 2. Else, proceed to Step 3.
3: If $\overrightarrow{\mathcal{A}}$ is identical at all the vertices of Θ_c^h and an infeasible combination, let p be a point in Θ_c^h. Else, proceed to Step 5.
4: If a p exists such that Problem P1 is feasible, then $p_c \leftarrow p$. Else, eliminate Θ_c^h from the parameter search space and terminate Algorithm 2.
5: Assign the center of mass of the vertex points to p_c. Terminate Algorithm 2.

Theorem 3. (Convex combination [248]) *The convex hull of set S, $H(S)$ is defined as the set of*

all convex combinations of S. Then $p \in H(S)$ if and only if p can be represented as follows.

$$\begin{aligned} p &= \sum_{i=1}^{r} \mu_i p_i \\ \sum_{i=1}^{r} \mu_i &= 1 \\ \mu_i \geq 0, p_i &\in S \end{aligned} \tag{3}$$

where r is the cardinality of point set S.

We know that there exists at least one critical region if Problem P1 can yield a feasible solution for $p \in H(\overrightarrow{\theta}_k^h)$. One can simply formulate an LP problem with arbitrary weights on the optimization variables and parameters, subject to the constraint set $g(x, \theta)$ and Eq. 3. However, the solution of this problem may return a point on a facet of Θ_c^h, provided the constructed LP is non-degenerate. Although any $p \in H(\overrightarrow{\theta}_k^h)$ is suitable to be assigned as p_c, selecting a point on a facet reduces the dimensionality of the search space to $n - 1$, ergo increases the number of triangulations in the future iterations.

Therefore, the feasibility problem is addressed by finding the Chebyshev center[3] of the constraint set $g(x, \theta)$ and Eq. 3. The Chebyshev center ensures that the located point p_c (i) is feasible for Problem P1, (ii) belongs to Θ_c^h, and (iii) does not lie on a facet of Θ_c^h. Note that finding the Chebyshev center can be replaced by any technique that finds a feasible point in Θ_c^h.

6.3.4 Handling degeneracy

The discussion hitherto has focused on non-degenerate parametric problems. However, degeneracy in multiparametric optimization problems has been reported as a significant complication that needs to addressed [250]. Two types of degeneracy are encountered in the literature, namely primal and dual.

[3]The Chebyshev center is defined as the center of the largest "ball" that can fit in a polytope. The interested reader is refered to Boyd and Vandenberghe [249] for details regarding the Chebyshev center.

6.3.4.1 Primal degeneracy

Let $\mathcal{A}_1$ and $\mathcal{A}_2$ be the active sets of two adjacent critical regions CR_1 and CR_2, respectively. Then, the active set at $\bar{\theta} = \{\theta \mid \theta \in CR_1 \cap CR_2\}$ is $\mathcal{A}_1 \cup \mathcal{A}_2$. If the rows of $A_{\mathcal{A}_1 \cup \mathcal{A}_2}$ are linearly dependent, the LICQ conditions are violated at $\bar{\theta}$. Problem P1 is primal degenerate at such conditions. A detailed discussion on primal degeneracy is provided by Tøndel et al. [236].

In the proposed algorithm, we address the primal degeneracy by perturbing the point of exploration p_c in a random direction such that the new point remains in the parent candidate simplex (i.e. $p'_c \in \Theta^h_c$). The perturbed point is replaced with the original point in all sibling subsets.

6.3.4.2 Dual degeneracy

Let $\mathcal{A}$ be the active set of Problem P1 at an arbitrary parameter realization $\bar{\theta}$. If there exists any $j \in \mathcal{A}$ such that $g_j = 0$ and $\lambda_j = 0$ (weakly active constraints), then the SCS condition is violated, and Problem P1 is dual degenerate at $\bar{\theta}$. Note that dual degeneracy can occur in mpLP problems, whereas the mpQP problem is guaranteed to have a unique solution to its dual counterpart in the feasible parameter space, since Problem P1 is defined as strictly convex ($Q \succ 0$). Therefore, the remaining discussion in this subsection focuses on dual degeneracy in mpLP problems.

Various strategies have been proposed in the literature to address the dual degeneracy problem [251, 250]. In this study, we follow a procedure summarized in Algorithm 3.

Algorithm 3 *Handling dual degeneracy in mpLP*

1: Get Problem P1, $\overrightarrow{\mathcal{A}_j}$, p_c.
2: Assign $\overrightarrow{\mathcal{A}}_j$ as the strongly active constraint set at p_c.
3: If $|\overrightarrow{\mathcal{A}}_j|$ is equal to the number of optimization variables, n, Problem P1 has a uniquely defined critical region at p_c, and return $\overrightarrow{\mathcal{A}}_j$.
4: If $|\overrightarrow{\mathcal{A}}_j|$ is less than the number of optimization variables, n, Problem P1 has overlapping critical regions at p_c. Then, determine a full rank A by selecting constraints among the weakly active constraint set, denoted as $\overrightarrow{\mathcal{A}}'_j$. Return the updated $\overrightarrow{\mathcal{A}}_j$.

The goal of Algorithm 3 is to force the LICQ condition to hold when it fails, by considering all of the proper combinations of the weakly active constraints. The cardinality of an active set combination, $|\overrightarrow{\mathcal{A}}_j|$, gives the number of optimization variables that can be uniquely determined for an mpLP problem at an arbitrary parameter realization[4]. Therefore, we know that the active set combination $\overrightarrow{\mathcal{A}}_j$ is dual degenerate at point p_c if the number of optimization variables exceeds the number of strongly active constraints.

If dual degeneracy exists, the pivot columns of $A_{\overrightarrow{\mathcal{A}}_j}$ indicate the variables that can be uniquely determined, and their complement yields the index of degenerate variables. Hence, we need to select $n - |\overrightarrow{\mathcal{A}}_j|$ linearly independent rows in $A_{\overrightarrow{\mathcal{A}}'_j}$ such that they are (i) orthogonal to $A_{\overrightarrow{\mathcal{A}}_j}$, and (ii) weakly active at the parameter realization. Determining such constraints defines a critical region.

Note that the number of weakly active constraints can exceed the required number of rows to force the LICQ conditions. In that case, all possible combinations that yield a full rank $A_{\overrightarrow{\mathcal{A}}_j}$ should be considered. However, each combination will yield overlapping critical regions.

Remark 6. Algorithm 3 allows for a separation between the space exploration and the dual degeneracy checking steps. A significant benefit of complete separation of these two steps is that the termination of the exploration algorithm is achieved regardless of the potential dual degeneracies in the solution space.

6.3.5 Extension to mixed-integer problems

The discussion thus far considers strictly continuous variables in the parametric problem P1. In this section, we extend the application of the algorithm to mpMILP, given by Problem 2 (P2).

$$\begin{aligned} z^*(\theta) = \quad & \min_{\omega} \quad c^T\omega \\ & s.t. \quad [A\ E]\omega \leq b + F\theta, \\ & \qquad \omega = [x^T\ y^T]^T,\ \theta \in \Theta \end{aligned} \tag{P2}$$

where $y \in \{0,1\}^p$, and all the matrices are of appropriate dimensions. The most significant

[4]Recall $\overrightarrow{\mathcal{A}}$ only includes the strongly active set.

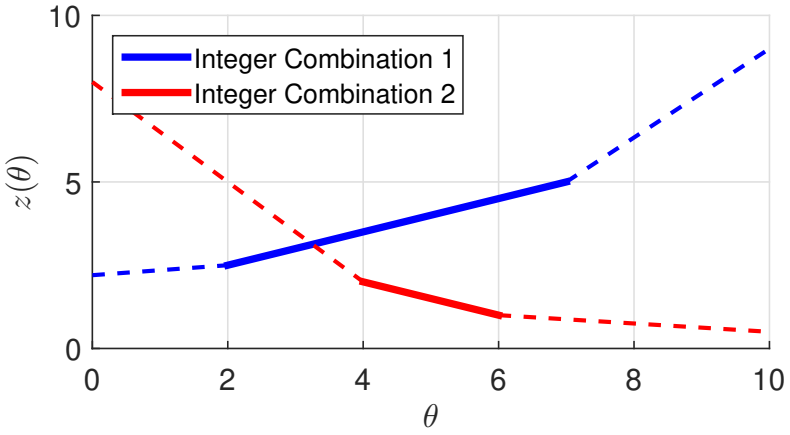


Figure 6.3: The map of critical regions with respect to the objective function for an arbitrary mpMILP problem with one parameter. Each color represents a different combination of integer variables. Notice that integer combination 1 gives a lower objective value at $\theta \in [2, 3.25]$. If the algorithm first samples a point within this range, the solid red critical region will never be explored since it is enclosed by the solid blue critical region.

challenge in this class of problems is the non-convexity of the feasible parameter space. Nonetheless, the elimination procedure described in Algorithm 2 can handle the non-convexity of the infeasible parameter space, rendering it possible to solve the class of problems described by Problem P2.

The solution returned by the proposed algorithm will span the entirety of the feasible parameter space Θ_f, however the optimality of the parametric expression across a critical region cannot be guaranteed. The reason for the loss of optimality stems from the overlapping layers of critical regions for every combination of binary variables. The overlap may result into overlooking the optimal critical regions in the existence of previously explored enclosing critical regions, as illustrated on a one dimensional mpMILP problem in Figure 6.3.

Hence, we propose a post-processing procedure that evaluates the explored critical regions and returns nonoverlapping optimal piecewise affine unique partitions, as outlined by Algorithm 4. For each explored critical region CR_j, we first exclude the optimal combination of binary variables by an integer cut, defined by Eq. 8 [252].

$$\sum_{i\in\mathcal{Y}} y_i + \sum_{i\in\mathcal{Y}'} y_i \leq |\mathcal{Y}| - 1,$$
$$\mathcal{Y} = \{i|y_i = 1\} \tag{8}$$

If P2 has a better solution in CR_j after excluding the existing binary variable combination, the optimal critical region is constructed around the new solution by Theorem 1, and denoted by CR_{new}. The remainder of CR_j is dissected into a set of polytopes, CR'_j. Each polytope in CR'_j is described by the space enclosed by CR_j, and the complement of a half plane that describes the new critical region CR_{new}. The set of polytopes CR'_j are further subjected to integer cuts in the subsequent iterations. The procedure is iterated until no feasible solution is found after including the integer cut.

Algorithm 4 *Postprocessing mpMILP solutions*

1: Get Problem P2 and the set of critical regions CR determined by Algorithm 1.
2: Let $g^{ic}(\omega, \theta)$ be the integer cut to the binary combination in CR_j.
3: If there exists a feasible parameter realization $\overline{\theta}$ in the polytope described by $g \cap g^{ic}$, find the active set of the optimal solution.
4: Construct the new critical region, CR_{new}, based on Theorem 1. Return the critical region as the optimal partition.
5: Define CR'_j as the set of polytopes that comprises the relative complement of CR_{new} in CR_j. Note that each of these polytopes is described by the space enclosed between the critical region CR_j and the complement of the hyperplanes that bound CR_{new}.
6: Add CR'_j to the set of critical regions CR. Increment j. Go back to Step 2.
7: Terminate when the set of critical regions CR is empty.

6.3.6 Limitations

While the proposed algorithm is effective in solving multiparametric problems with a large number of optimization variables and constraints, it suffers handling large number of parameters. This limitation is a direct consequence of the triangulation step, where determining the non-empty and non-overlapping candidate simplices is computationally taxing. The primary difficulty arises from the triangulation in the first iteration, where the algorithm generates $\binom{2^q+1}{q+1}$ child simplices,

while the remaining iterations the number reduces to $\binom{q+2}{q+1}$, i.e. $q+2$. The reason for the sharp decrease is that the first triangulation step takes place in a q dimensional hypercube, which has 2^q vertex points[5]. On the other hand, a simplex has $q+1$ vertex points, which enables the triangulation in significantly higher dimensions. Based on this fact, our current research focuses on alleviating the computational burden by constructing the tightest overarching simplex to initialize the triangulation in higher dimensions.

Additionally, further improvement can be achieved by developing stronger termination criteria. For instance, the facet-to-facet property [237] can be introduced to the proposed algorithm to avoid redundant triangulations between two adjacent critical regions that are already explored.

6.4 Numerical Examples

6.4.1 Performance against state-of-the-art solvers

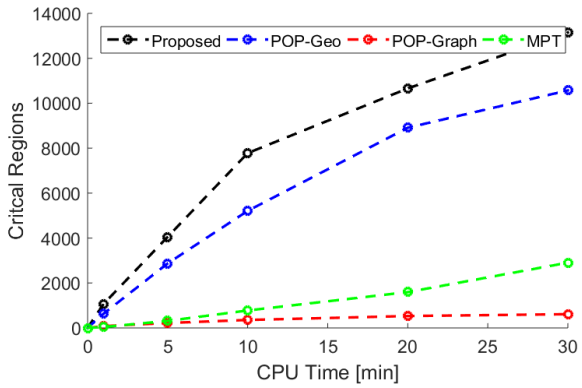
The proposed algorithm is compared against state-of-the-art multiparametric solvers that can be found in the POP toolbox [253] and the Multiparametric toolbox (MPT) [254]. In the POP toolbox, the solvers used are the connected-graph and geometrical, which will be referred to as POP-Graph and POP-Geo, respectively. The MPT solver used is the mpQP algorithm for the mpLP and mpQP. MPT does not maintain a solver for mpMILP problems, and therefore is not considered for this problem class. The numerical example problems used as a basis of comparison for the different algorithms are randomly generated and summarized in Table 6.1. All of the experiments were done on a 4 core machine using an Intel i7-4770 CPU at 3.40 GHz and 16 GB of RAM. Additionally, the tests were run using the MATLAB environment.

The three problems generated for comparison are large in size and are described in Table 6.1, where n is the number of optimization variables, q is the number of parameters, m is the number of constraints, and p is the number of binary variables. Determining the full solution to the problems generated requires a significant amount of time, therefore the algorithms tested were allotted 30 minutes to explore the solution space. The results for the number of critical regions identified for each algorithm at the intervals of 1, 5, 10, 20, and 30 minutes are provided in Figures 6.4a,

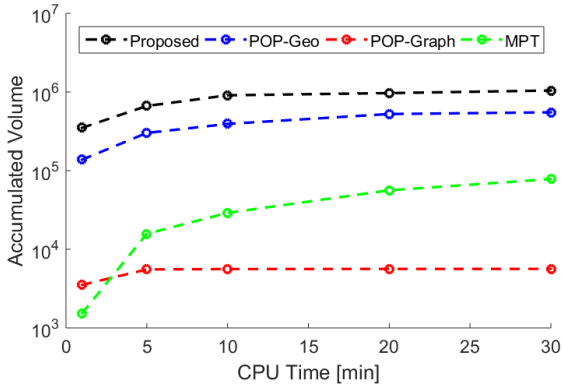
[5]In most practical applications, Θ is usually described by box constraints, which yield 2^q vertex points.

Table 6.1: Details for the mpLP, mpQP, and mpMILP problems.

	n	p	q	m
mpLP	30	N/A	5	70
mpQP	100	N/A	5	150
mpMILP	20	10	5	50



(a) Number of critical regions identified for each algorithm for the mpLP.

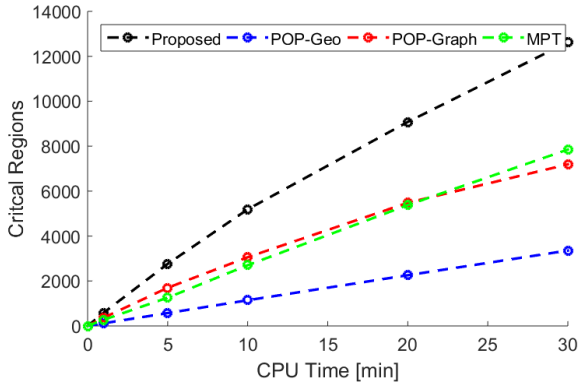


(b) Volume of the agglomeration of critical regions for the mpLP.

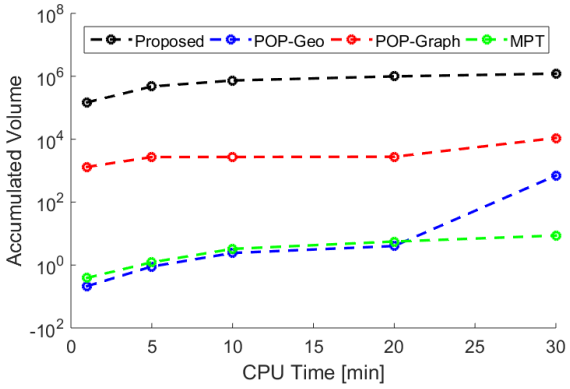
Figure 6.4: Comparisons of the tested algorithms on the mpLP.

6.5a, and 6.6a. For the mpLP and mpQP problems, the proposed strategy is able to identify the largest number of critical regions for the entire 30 minute duration. However, for the mpMILP problem, the proposed approach has identified significantly less critical regions than POP-Geo and POP-Graph.

The algorithm identifying a large number of critical regions is promising, however, a more promising result lies in Figures 6.4b, 6.5b, and 6.6b. These figures provide details for the total volume occupied by the identified critical regions at the specified interval. In all of the problems considered, the proposed strategy identified the critical regions associated with the largest volume. This phenomenon is a result of the sample based strategy the proposed algorithm employs; a sampled point from the parameter space is more likely to exist in a larger critical region, and thus be identified.

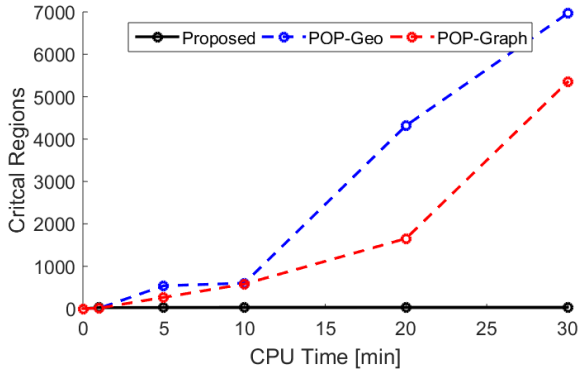


(a) Number of critical regions identified for each algorithm for the mpQP.

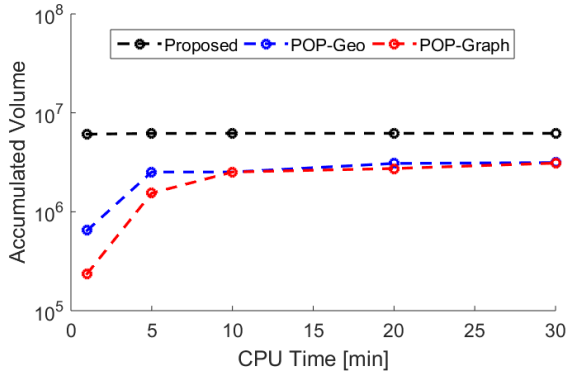


(b) Volume of the agglomeration of critical regions for the mpQP.

Figure 6.5: Comparisons of the tested algorithms on the mpQP.



(a) Number of critical regions identified for each algorithm for the mpMILP.



(b) Volume of the agglomeration of critical regions for the mpMILP.

Figure 6.6: Comparisons of the tested algorithms on the mpMILP.

Table 6.2: Summary of results for accumulated volume example.

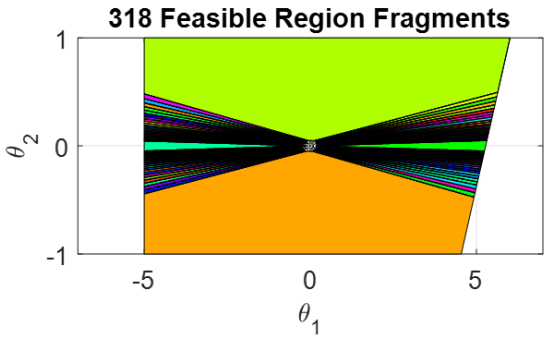
	Proposed	POP-Geo	POP-Graph	MPT
critical regions	318	1358	3358	3340
accumulated volume (m.u.)	20.52	18.72	0.02	0.02

6.4.2 Accumulated volume analysis

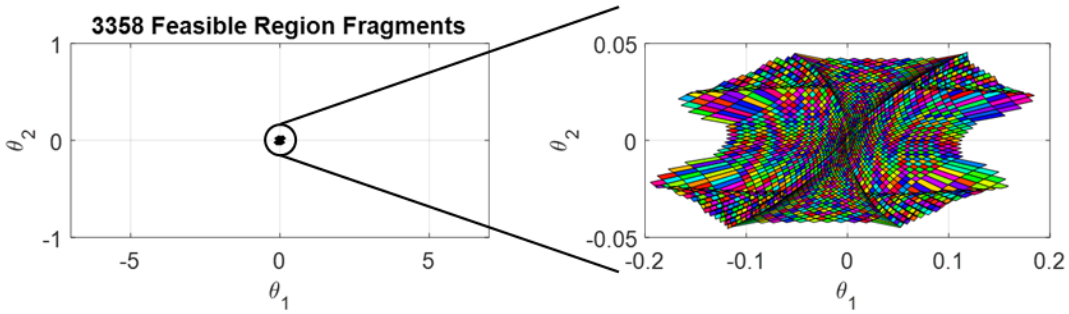
To illustrate the concept of accumulated volume, an mpQP is generated with two parameters for visualization. The problem has 800 constraints and 50 optimization variables. Each algorithm is allotted 5 minutes, and the number of identified critical regions and accumulated volume are detailed in Table 6.2. From Table 6.2, it is evident that POP-Graph identified the most critical regions, and the proposed algorithm determined the largest accumulated volume. Therefore, a visual comparison is made between the proposed approach and POP-Graph to highlight the difference in concepts of number of critical regions identified and explored parameter space volume. The comparison is seen in Figure 6.7, where POP-Graph identified 3358 critical regions and the proposed algorithm determined 318 critical regions. Plotting the map of solutions on the same scale, the critical regions identified by the proposed algorithm occupy a significantly larger volume of Θ_f compared to the critical regions identified by POP-Graph. To summarize, POP-Graph is able to identify over ten times the number of critical regions, but the developed solution for the proposed approach accounts for 1000 times more accumulated volume. This comparison showcases the proposed strategy prioritizes identifying larger critical regions and the importance for a method to analyze partial multiparametric solutions that does not rely on the number of critical regions explored. By analyzing the accumulated volume of the partial solution, a better representation of the developed solution is determined, because the likelihood a parameter realization will exist in an explored critical region is directly correlated with the total volume identified.

6.5 Conclusions

In this work, a novel space exploration was presented for mpLP, mpQP, and mpMILP problems. We employed Delaunay triangulation to effectively partition the parameter space into non-



(a) The identified critical regions from the proposed algorithm.



(b) The identified critical regions from POP-Graph, where the right figure is a zoomed in version of the left figure.

Figure 6.7: Map of solutions for the proposed algorithm and POP-Graph.

increasing sets as simplices. These simplices were eliminated from the parameter search space and excluded for further partitioning when determined to be a subset of an optimal unique partition. Another contribution of this paper is investigating the volume occupied by the explored critical regions. Due to the nature of the proposed exploration strategy, critical regions that occupy a larger portion of volume in the parameter space are prioritized. The ability to prioritize larger critical regions is a salient feature that is not exhibited by existing multiparametric algorithms. We presented randomly generated numerical examples that are large scale in the number of variables and constraints to showcase the proposed algorithms ability to identify the volumetrically larger critical regions compared to the state-of-the-art solvers, especially in the early phase of the exploration. Furthermore, we presented a simplistic procedure for the degenerate cases for the sake of completeness. However, handling degeneracy is a major challenge in multiparametric programming problems, and thus developing more efficient algorithms for degenerate problems is a future direction.

The developed algorithm is well suited for large scale multi-level optimization problems, where determining a feasible solution is challenging. In these multi-level optimization problems where multiparametric programming provides the offline explicit expressions for operations decisions, such as scheduling and control, the integrated optimization formulation is intractable for practical applications in which the operational optimization problems are large. The proposed algorithm, by prioritizing the large partitions of the solution space, allows for a tractable formulation for these integrated optimization problems, and the identified critical regions provide a feasible overestimator to the global solution while maintaining tractability. Moreover, the exploration procedure allows for efficient parallelization due to the complete independence of the calculation cycles, which is another significant benefit to address large scale problems.

The promising results shown in this work encourages future research utilizing the underlying concepts of the proposed algorithm. Therefore, current research focuses on (i) improving scalability with the number of parameters, and (ii) developing stronger termination criteria, as well as (iii) extending the applicability of the algorithm to multiparametric nonlinear programming problems.

7. CONCLUSIONS

The aim of this thesis was to develop a unified, process agnostic strategy to integrate the process design and operational optimization decisions, namely scheduling and control, using a single high fidelity model. The main benefits of the proposed simultaneous approach can be summarized in two folds. First, optimality of the overall dynamic system was improved significantly by leveraging the synergistic interactions between the individual decision layers. Conventional approaches isolate these layers hierarchically and sequentially to achieve simplified models due to lack of a coherent modeling technique that simultaneously accounts for their interactions. Neglecting these interactions results in suboptimal decisions. Second, direct inclusion of advanced control dynamics in the integrated problem allowed for providing certificates of operability in closed loop implementation. Since the longer time scale decisions are aware of the exact dynamics dictated by the faster decisions, they can impose realizable targets in closed loop.

The proposed methodology relies on developing explicit expressions for the optimal scheduling and control strategies as a function of the process design variables via multi-parametric programming. These design dependent functions are included in the design optimization problem, which is posed as a mixed integer dynamic optimization formulation. Therefore, the overall degrees of freedom of the integrated problem is reduced to the design variables only, which decreases the feasible search space.

The proposed integration approach was first illustrated on a continuous binary distillation column for the separation of toluene and benzene. Scheduling decisions were excluded in this study, since the goal of the process is to track time-independent set points. Integration of scheduling and control decisions was showcased on a multiproduct CSTR, and two CSTRs operating in parallel. The complete framework was then demonstrated on a residential scale combined heat and power system, which supplies electricity and hot water to a residential district in parallel to the utilities. Finally, a batch process with multiple tasks was considered, which required the use of a more complex representation, namely State Equipment Network, in order to systematically handle

task sequencing, timing, and resource allocation. One challenge to embed the explicit optimal control law into the mixed integer dynamic optimization problem was the size of the offline solutions, which was addressed by introducing a base-2 numeral system based modeling technique to exponentially reduce the required number of binary variables

7.1 Summary of the key contributions

The key benefits and observations of the proposed approach is listed as follows.

(i) Optimality of the decisions was improved significantly compared to conventional approaches. This improvement is achieved by formulating an enlarged search space that includes the interactions between decision layers.

(ii) Direct implementation of the faster time operational scale decisions in longer time strategic decisions allow for certificates of operability in closed loop implementation.

(iii) Sizes of the processing equipment was reduced, indicating that we can achieve the benefits of process intensification through integrating the temporal decisions.

(iv) Operating costs were decreased through reduced raw material and energy utilization.

(v) Offline maps of explicit optimal operations at both long and short terms allow for much shorter online computational time. Therefore, the proposed framework is applicable to processes with fast dynamics. In addition, these offline maps can be utilized to make strategic decisions prior to the operation, such as exploring the range of feasible operation in the existence of process faults, as presented by Onel et al. (2020) [229].

(vi) The complete list of publications that feature these contributions is given in Appendix C.

7.2 Recommendations for future work

Multi-parametric programming and explicit model predictive control are at the heart of the proposed framework. However, the size of the explicit solutions is a limitation to solve the integrated

dynamic optimization problem. Therefore, we proposed a novel algorithm for large scale multi-parametric programs, where the exploration algorithm prioritizes the volumetrically large critical regions. The computational experiments demonstrated that the proposed algorithm explores the parameter space orders of magnitude faster in volume compared to the existing state-of-the-art solvers. One of the future efforts will focus on using this algorithm to explore the parameter space of the model predictive control and scheduling problems. A partial solution acquired by this algorithm can be embedded in the integrated dynamic optimization problem, the solution of which will yield an overestimator to the complete problem. Therefore, a feasible solution for all the decisions can be derived simultaneously with a guarantee of operability in closed loop.

Uncertainty assessment is another direction of our future work. The current approach handles the design variables in the scheduling and control levels as right hand side parameters in the multi-parametric programming formulations. However, this approach assumes linear interactions between the design variables and the operational strategies. Robust counterparts of these strategies will capture the effects of the design variables with higher accuracy. However, this will result in a multi-parametric program with left hand side uncertainty, for which a rigorous solution methodology is not yet available. Our current efforts focus on developing an algorithm to solve this class of optimization problems, which can improve the effectiveness of the proposed integrated approach.

Further improvement can be achieved by incorporating more complex process control schemes. The proposed approach employed linear model predictive control (MPC) due the availability of an exact offline solution. Linear MPC utilizes a linear state space model, which restricts the capacity of the controller to handle nonlinear relations in the control level. Although nonlinear MPC (NMPC) can deliver an improved closed loop performance, a generalized strategy to derive its multi-parametric counterpart quite challenging. Our recent efforts focus on developing solution strategies for specific class of problems, namely multi-parametric quadratically constrained quadratic programming problems, which can be implemented in the integrated design formulation [255].

Finally, the system measurements may not always be reliable due to faults that can rise in the

process instrumentation. Such faults will result in deviations from the desired operating points. In these cases, a fault tolerant control strategy should be adopted in the integrated dynamic optimization formulation to proactively account for such process upsets. In Onel et al. (2020) [229], we developed a multi-parametric fault tolerant model predictive control scheme that maps the optimal control actions as a function of the potential process upsets. This approach allows for the detection of the feasible range of operation in the existence of process faults. Our future work will include embedding this proposed control strategy in the integrated design problem to yield more reliable closed strategies with known type process faults. The interested user can appeal to Appendix B for further details.

REFERENCES

[1] I. Grossmann, "Enterprise-wide optimization: A new frontier in process systems engineering," *AIChE Journal*, vol. 51, no. 7, pp. 1846–1857, 2005.

[2] E. N. Pistikopoulos and N. A. Diangelakis, "Towards the integration of process design, control and scheduling: Are we getting closer?," *Computers & Chemical Engineering*, vol. 91, pp. 85 – 92, 2016.

[3] R. W. H. Sargent, "Integrated design and optimization of processes," *Chemical Engineering Progress*, vol. 63, no. 9, pp. 71 – 78, 1967.

[4] S. Terrazas-Moreno, A. Flores-Tlacuahuac, and I. E. Grossmann, "Simultaneous design, scheduling, and optimal control of a methyl-methacrylate continuous polymerization reactor," *AIChE Journal*, vol. 54, no. 12, pp. 3160–3170, 2008.

[5] B. P. Patil, E. Maia, and L. A. Ricardez-sandoval, "Integration of Scheduling, Design, and Control of Multiproduct Chemical Processes Under Uncertainty," *AIChE Journal*, vol. 61, no. 8, 2015.

[6] R. W. Koller and L. A. Ricardez-Sandoval, "A dynamic optimization framework for integration of design, control and scheduling of multi-product chemical processes under disturbance and uncertainty," *Computers & Chemical Engineering*, vol. 106, pp. 147 – 159, 2017. ESCAPE-26.

[7] R. W. Koller, L. A. Ricardez-Sandoval, and L. T. Biegler, "Stochastic back-off algorithm for simultaneous design, control, and scheduling of multiproduct systems under uncertainty," *AIChE Journal*, vol. 64, no. 7, pp. 2379–2389, 2018.

[8] B. Burnak, N. A. Diangelakis, J. Katz, and E. N. Pistikopoulos, "Integrated process design, scheduling, and control using multiparametric programming," *Computers & Chemical Engineering*, 2019.

[9] B. Burnak, N. A. Diangelakis, and E. N. Pistikopoulos, "Towards the grand unification of process design, scheduling, and control—utopia or reality?," *Processes*, vol. 7, no. 7, 2019.

[10] T. Takamatsu, I. Hashimoto, and H. Ohno, "Optimal design of a large complex system from the viewpoint of sensitivity analysis," *Industrial & Engineering Chemistry Process Design and Development*, vol. 9, no. 3, pp. 368–379, 1970.

[11] N. Nishida, A. Ichikawa, and E. Tazaki, "Synthesis of optimal process systems with uncertainty," *Industrial & Engineering Chemistry Process Design and Development*, vol. 13, no. 3, pp. 209–214, 1974.

[12] I. E. Grossmann and R. W. H. Sargent, "Optimum design of chemical plants with uncertain parameters," *AIChE Journal*, vol. 24, no. 6, pp. 1021–1028, 1978.

[13] B. M. Kwak and E. J. Haug, "Optimum design in the presence of parametric uncertainty," *Journal of Optimization Theory and Applications*, vol. 19, pp. 527–546, Aug 1976.

[14] K. P. Halemane and I. E. Grossmann, "Optimal process design under uncertainty," *AIChE Journal*, vol. 29, pp. 425–433, 5 1983.

[15] R. E. Swaney and I. E. Grossmann, "An index for operational flexibility in chemical process design. part i: Formulation and theory," *AIChE Journal*, vol. 31, pp. 621–630, 4 1985.

[16] R. E. Swaney and I. E. Grossmann, "An index for operational flexibility in chemical process design. part ii: Computational algorithms," *AIChE Journal*, vol. 31, pp. 631–641, 4 1985.

[17] I. E. Grossmann and C. A. Floudas, "Active constraint strategy for flexibility analysis in chemical processes," *Computers & Chemical Engineering*, vol. 11, no. 6, pp. 675 – 693, 1987.

[18] C. A. Floudas and I. E. Grossmann, "Synthesis of flexible heat exchanger networks with uncertain flowrates and temperatures," *Computers & Chemical Engineering*, vol. 11, no. 4, pp. 319 – 336, 1987.

[19] Y. Shimizu, "A plain approach for dealing with flexibility problems in linear systems," *Computers & Chemical Engineering*, vol. 13, no. 10, pp. 1189 – 1191, 1989.

[20] Y. Shimizu, "Application of flexibility analysis for compromise solution in large-scale linear systems," *Journal of Chemical Engineering of Japan*, vol. 22, no. 2, pp. 189–194, 1989.

[21] V. Bansal, J. D. Perkins, and E. N. Pistikopoulos, "Flexibility analysis and design of linear systems by parametric programming," *AIChE Journal*, vol. 46, pp. 335–354, 2 2000.

[22] E. N. Pistikopoulos and I. E. Grossmann, "Optimal retrofit design for improving process flexibility in linear systems," *Computers & Chemical Engineering*, vol. 12, no. 7, pp. 719 – 731, 1988. Special Issue on Process Systems Engineering.

[23] E. N. Pistikopoulos and I. E. Grossmann, "Stochastic optimization of flexibility in retrofit design of linear systems," *Computers & Chemical Engineering*, vol. 12, no. 12, pp. 1215 – 1227, 1988.

[24] E. N. Pistikopoulos and I. E. Grossmann, "Optimal retrofit design for improving process flexibility in nonlinear systems—i. fixed degree of flexibility," *Computers & Chemical Engineering*, vol. 13, no. 9, pp. 1003 – 1016, 1989.

[25] E. N. Pistikopoulos and I. E. Grossmann, "Optimal retrofit design for improving process flexibility in nonlinear systems—ii. optimal level of flexibility," *Computers & Chemical Engineering*, vol. 13, no. 10, pp. 1087 – 1096, 1989.

[26] E. N. Pistikopoulos and I. E. Grossmann, "Evaluation and redesign for improving flexibility in linear systems with infeasible nominal conditions," *Computers & Chemical Engineering*, vol. 12, no. 8, pp. 841 – 843, 1988.

[27] C. Raspanti, J. Bandoni, and L. Biegler, "New strategies for flexibility analysis and design under uncertainty," *Computers & Chemical Engineering*, vol. 24, no. 9, pp. 2193 – 2209, 2000.

[28] G. Kreisselmeier and R. Steinhauser, "Systematic control design by optimizing a vector performance index," in *Computer Aided Design of Control Systems* (M. CUENOD, ed.), pp. 113 – 117, Pergamon, 1980.

[29] C. Chen and O. L. Mangasarian, "A class of smoothing functions for nonlinear and mixed complementarity problems," *Computational Optimization and Applications*, vol. 5, pp. 97–138, Mar 1996.

[30] E. N. Pistikopoulos and T. A. Mazzuchi, "A novel flexibility analysis approach for processes with stochastic parameters," *Computers & Chemical Engineering*, vol. 14, no. 9, pp. 991 – 1000, 1990.

[31] D. A. Straub and I. E. Grossmann, "Integrated stochastic metric of flexibility for systems with discrete state and continuous parameter uncertainties," *Computers & Chemical Engineering*, vol. 14, no. 9, pp. 967 – 985, 1990.

[32] D. A. Straub and I. E. Grossmann, "Design optimization of stochastic flexibility," *Computers & Chemical Engineering*, vol. 17, no. 4, pp. 339 – 354, 1993. An International Journal of Computer Applications in Chemical Engineering.

[33] V. D. Dimitriadis and E. N. Pistikopoulos, "Flexibility analysis of dynamic systems," *Industrial & Engineering Chemistry Research*, vol. 34, no. 12, pp. 4451–4462, 1995.

[34] H. Zhou, X. Li, Y. Qian, Y. Chen, and A. Kraslawski, "Optimizing the initial conditions to improve the dynamic flexibility of batch processes," *Industrial & Engineering Chemistry Research*, vol. 48, no. 13, pp. 6321–6326, 2009.

[35] M. J. Mohideen, J. D. Perkins, and E. N. Pistikopoulos, "Optimal design of dynamic systems under uncertainty," *AIChE Journal*, vol. 42, pp. 2251–2272, 8 1996.

[36] M. J. Mohideen, J. D. Perkins, and E. N. Pistikopoulos, "Optimal synthesis and design of dynamic systems under uncertainty," *Computers & Chemical Engineering*, vol. 20, pp. S895 – S900, 1996. European Symposium on Computer Aided Process Engineering-6.

[37] M. J. Mohideen, J. D. Perkins, and E. N. Pistikopoulos, "Robust stability considerations in optimal design of dynamic systems under uncertainty," *Journal of Process Control*, vol. 7, no. 5, pp. 371 – 385, 1997. Nonlinear Process Control.

[38] A. D. Pretoro, L. Montastruc, F. Manenti, and X. Joulia, "Flexibility analysis of a distillation column: Indexes comparison and economic assessment," *Computers & Chemical Engineering*, vol. 124, pp. 93 – 108, 2019.

[39] Y. Zhu, S. Legg, and C. D. Laird, "Optimal design of cryogenic air separation columns under uncertainty," *Computers & Chemical Engineering*, vol. 34, no. 9, pp. 1377 – 1384, 2010. Selected papers from the 7th International Conference on the Foundations of Computer-Aided Process Design (FOCAPD, 2009, Breckenridge, Colorado, USA.

[40] W. Huang, X. Li, S. Yang, and Y. Qian, "Dynamic flexibility analysis of chemical reaction systems with time delay: Using a modified finite element collocation method," *Chemical Engineering Research and Design*, vol. 89, no. 10, pp. 1938 – 1946, 2011.

[41] A. E. S. Konukman, M. C. Çamurdan, and U. Akman, "Simultaneous flexibility targeting and synthesis of minimum-utility heat-exchanger networks with superstructure-based milp formulation," *Chemical Engineering and Processing: Process Intensification*, vol. 41, no. 6, pp. 501 – 518, 2002.

[42] A. E. S. Konukman and U. Akman, "Flexibility and operability analysis of a hen-integrated natural gas expander plant," *Chemical Engineering Science*, vol. 60, no. 24, pp. 7057 – 7074, 2005.

[43] M. Escobar, J. O. Trierweiler, and I. E. Grossmann, "Simultaneous synthesis of heat exchanger networks with operability considerations: Flexibility and controllability," *Computers & Chemical Engineering*, vol. 55, pp. 158 – 180, 2013.

[44] D. K. Varvarezos, I. E. Grossmann, and L. T. Biegler, "An outer-approximation method for multiperiod design optimization," *Industrial & Engineering Chemistry Research*, vol. 31, no. 6, pp. 1466–1477, 1992.

[45] E. Pistikopoulos and M. Ierapetritou, "Novel approach for optimal process design under uncertainty," *Computers & Chemical Engineering*, vol. 19, no. 10, pp. 1089 – 1110, 1995.

[46] M. Morari, "Design of resilient processing plants—iii: A general framework for the assessment of dynamic resilience," *Chemical Engineering Science*, vol. 38, no. 11, pp. 1881 – 1891, 1983.

[47] M. Morari, "Flexibility and resiliency of process systems," *Computers & Chemical Engineering*, vol. 7, no. 4, pp. 423 – 437, 1983.

[48] I. E. Grossmann and M. Morari, "Operability, Resiliency, and Flexibility 2: process design objectives for a changing world," 1 1983.

[49] M. Morari, W. Grimm, M. J. Oglesby, and I. D. Prosser, "Design of resilient processing plants—vii. design of energy management system for unstable reactors—new insights," *Chemical Engineering Science*, vol. 40, no. 2, pp. 187 – 198, 1985.

[50] A. Palazoglu, B. Manousiouthakis, and Y. Arkun, "Design of chemical plants with improved dynamic operability in an environment of uncertainty," *Industrial & Engineering Chemistry Process Design and Development*, vol. 24, no. 3, pp. 802–813, 1985.

[51] A. Palazoglu and Y. Arkun, "A multiobjective approach to design chemical plants with robust dynamic operability characteristics," *Computers & Chemical Engineering*, vol. 10, no. 6, pp. 567 – 575, 1986.

[52] S. Skogestad and M. Morari, "Design of resilient processing plants-ix. effect of model uncertainty on dynamic resilience," *Chemical Engineering Science*, vol. 42, no. 7, pp. 1765 – 1780, 1987.

[53] R. D. Colberg, M. Morari, and D. W. Townsend, "A resilience target for heat exchanger network synthesis," *Computers & Chemical Engineering*, vol. 13, no. 7, pp. 821 – 837, 1989.

[54] J. D. Perkins and M. P. F. Wong, "Assessing controllability of chemical plants," *Chemical Engineering Research and Design*, vol. 63, no. 6, pp. 358–362, 1985.

[55] H. H. Rosenbrock, *State-space and multivariable theory*. Studies in dynamical systems series, Wiley Interscience Division, 1970.

[56] P. Psarris and C. A. Floudas, "Improving dynamic operability in mimo systems with time delays," *Chemical Engineering Science*, vol. 45, no. 12, pp. 3505 – 3524, 1990.

[57] P. Psarris and C. A. Floudas, "Dynamic operability of mimo systems with time delays and transmission zeroes—i. assessment," *Chemical Engineering Science*, vol. 46, no. 10, pp. 2691 – 2707, 1991.

[58] P. Psarris and C. A. Floudas, "Dynamic operability of mimo systems with time delays and transmission zeroes—ii. enhancement," *Chemical Engineering Science*, vol. 46, no. 10, pp. 2709 – 2728, 1991.

[59] G. W. Barton, W. K. Chan, and J. D. Perkins, "Interaction between process design and process control: the role of open-loop indicators," *Journal of Process Control*, vol. 1, no. 3, pp. 161 – 170, 1991.

[60] L. Narraway, J. Perkins, and G. Barton, "Interaction between process design and process control: economic analysis of process dynamics," *Journal of Process Control*, vol. 1, no. 5, pp. 243 – 250, 1991.

[61] L. Narraway and J. Perkins, "Selection of process control structure based on economics," *Computers & Chemical Engineering*, vol. 18, pp. S511 – S515, 1994. European Symposium on Computer Aided Process Engineering—3.

[62] P. Bahri, J. Bandoni, G. Barton, and J. Romagnoli, "Back-off calculations in optimising control: A dynamic approach," *Computers & Chemical Engineering*, vol. 19, pp. 699 – 708, 1995.

[63] S. Walsh and J. Perkins, "Integrated design of effluent treatment systems," *IFAC Proceedings Volumes*, vol. 25, no. 24, pp. 107 – 112, 1992. IFAC Workshop on Interactions between Process Design and Process Control, London, UK, 7-8 September.

[64] M. L. Luyben and C. A. Floudas, "A multiobjective optimization approach for analyzing the interaction of design and control," *IFAC Proceedings Volumes*, vol. 25, no. 24, pp. 101 – 106, 1992. IFAC Workshop on Interactions between Process Design and Process Control, London, UK, 7-8 September.

[65] N. Shah, C. C. Pantelides, and R. W. H. Sargent, "The design and scheduling of multipurpose batch plants," *IFAC Proceedings Volumes*, vol. 25, no. 24, pp. 203 – 208, 1992. IFAC Workshop on Interactions between Process Design and Process Control, London, UK, 7-8 September.

[66] E. Kondili, C. Pantelides, and R. Sargent, "A general algorithm for short-term scheduling of batch operations—i. milp formulation," *Computers & Chemical Engineering*, vol. 17, no. 2, pp. 211 – 227, 1993. An International Journal of Computer Applications in Chemical Enginnering.

[67] T. V. Thomaidis and E. N. Pistikopoulos, "Design of flexible and reliable process systems," *IFAC Proceedings Volumes*, vol. 25, no. 24, pp. 235 – 240, 1992. IFAC Workshop on Interactions between Process Design and Process Control, London, UK, 7-8 September.

[68] S. Walsh and J. Perkins, "Application of integrated process and control system design to waste water neutralisation," *Computers & Chemical Engineering*, vol. 18, pp. S183 – S187, 1994. European Symposium on Computer Aided Process Engineering—3.

[69] L. T. Narraway and J. D. Perkins, "Selection of process control structure based on linear dynamic economics," *Industrial & Engineering Chemistry Research*, vol. 32, no. 11, pp. 2681–2692, 1993.

[70] M. L. Luyben and C. A. Floudas, "Analyzing the interaction of design and control—1. a multiobjective framework and application to binary distillation synthesis," *Computers & Chemical Engineering*, vol. 18, no. 10, pp. 933 – 969, 1994. An International Journal of Computer Application in Chemical Engineering.

[71] M. L. Luyben and C. A. Fluodas, "Analyzing the interaction of design and control—2. reactor-separator-recycle system," *Computers & Chemical Engineering*, vol. 18, no. 10, pp. 971 – 993, 1994. An International Journal of Computer Application in Chemical Engineering.

[72] V. Bansal, J. D. Perkins, E. . Pistikopoulos, R. Ross, and J. M. G. van Schijndel, "Simultaneous design and control optimisation under uncertainty," *Computers & Chemical Engineering*, vol. 24, no. 2, pp. 261 – 266, 2000.

[73] L. A. R. Sandoval, H. M. Budman, and P. L. Douglas, "Simultaneous design and control of processes under uncertainty: A robust modelling approach," *Journal of Process Control*, vol. 18, no. 7, pp. 735 – 752, 2008.

[74] L. A. Ricardez-Sandoval, H. M. Budman, and P. L. Douglas, "Application of robust control tools to the simultaneous design and control of dynamic systems," *Industrial & Engineering Chemistry Research*, vol. 48, no. 2, pp. 801–813, 2009.

[75] L. A. Ricardez-Sandoval, H. M. Budman, and P. L. Douglas, "Simultaneous design and control of chemical processes with application to the tennessee eastman process," *Journal of Process Control*, vol. 19, no. 8, pp. 1377 – 1391, 2009. Special Section on Hybrid Systems: Modeling, Simulation and Optimization.

[76] I. K. Kookos and J. D. Perkins, "Control structure selection based on economics: Generalization of the back-off methodology," *AIChE Journal*, vol. 62, no. 9, pp. 3056–3064, 2016.

[77] S. Mehta and L. A. Ricardez-Sandoval, "Integration of design and control of dynamic systems under uncertainty: A new back-off approach," *Industrial & Engineering Chemistry Research*, vol. 55, no. 2, pp. 485–498, 2016.

[78] M. Rafiei-Shishavan, S. Mehta, and L. A. Ricardez-Sandoval, "Simultaneous design and control under uncertainty: A back-off approach using power series expansions," *Computers & Chemical Engineering*, vol. 99, pp. 66 – 81, 2017.

[79] M. Rafiei and L. A. Ricardez-Sandoval, "Stochastic back-off approach for integration of design and control under uncertainty," *Industrial & Engineering Chemistry Research*, vol. 57, no. 12, pp. 4351–4365, 2018.

[80] I. K. Kookos and J. D. Perkins, "An algorithm for simultaneous process design and control," *Industrial & Engineering Chemistry Research*, vol. 40, no. 19, pp. 4079–4088, 2001.

[81] A. Malcolm, J. Polan, L. Zhang, B. A. Ogunnaike, and A. A. Linninger, "Integrating systems design and control using dynamic flexibility analysis," *AIChE Journal*, vol. 53, no. 8, pp. 2048–2061, 2007.

[82] J. Moon, S. Kim, and A. A. Linninger, "Integrated design and control under uncertainty: Embedded control optimization for plantwide processes," *Computers & Chemical Engineering*, vol. 35, no. 9, pp. 1718 – 1724, 2011. Energy Systems Engineering.

[83] S. Qin and T. A. Badgwell, "A survey of industrial model predictive control technology," *Control Engineering Practice*, vol. 11, no. 7, pp. 733 – 764, 2003.

[84] D. D. Brengel and W. D. Seider, "Coordinated design and control optimization of nonlinear processes," *Computers & Chemical Engineering*, vol. 16, no. 9, pp. 861 – 886, 1992. An International Journal of Computer Applications in Chemical Engineering.

[85] A. Bemporad, M. Morari, V. Dua, and E. N. Pistikopoulos, "The explicit linear quadratic regulator for constrained systems," *Automatica*, vol. 38, no. 1, pp. 3 – 20, 2002.

[86] V. Sakizlis, J. D. Perkins, and E. N. Pistikopoulos, "Parametric controllers in simultaneous process and control design optimization," *Industrial & Engineering Chemistry Research*, vol. 42, no. 20, pp. 4545–4563, 2003.

[87] N. A. Diangelakis and E. N. Pistikopoulos, "Model-based multi-parametric programming strategies towards the integration of design, control and operational optimization," in *27th European Symposium on Computer-Aided Process Engineering (ESCAPE-27)*, pp. 1867 – 1872, Elsevier, 2017.

[88] N. A. Diangelakis, B. Burnak, J. Katz, and E. N. Pistikopoulos, "Process design and control optimization: A simultaneous approach by multi-parametric programming," *AIChE Journal*, vol. 63, no. 11, pp. 4827–4846, 2017.

[89] K. B. Sanchez-Sanchez and L. A. Ricardez-Sandoval, "Simultaneous design and control under uncertainty using model predictive control," *Industrial & Engineering Chemistry Research*, vol. 52, no. 13, pp. 4815–4833, 2013.

[90] D. B. Birewar and I. E. Grossmann, "Incorporating scheduling in the optimal design of multiproduct batch plants," *Computers & Chemical Engineering*, vol. 13, no. 1, pp. 141 – 161, 1989. Computer Applications to Batch Chemical Processing.

[91] V. White, J. Perkins, and D. Espie, "Switchability analysis," *Computers & Chemical Engineering*, vol. 20, no. 4, pp. 469 – 474, 1996.

[92] T. Bhatia and L. Biegler, "Dynamic optimization in the design and scheduling of multiproduct batch plants," *Industrial & engineering chemistry research*, vol. 5885, no. 95, pp. 2234–2246, 1996.

[93] T. K. Bhatia and L. T. Biegler, "Dynamic optimization for batch design and scheduling with process model uncertainty," *Industrial and Engineering Chemistry Research*, vol. 36, no. 9, pp. 3708–3717, 1997.

[94] M. Baldea and I. Harjunkoski, "Integrated production scheduling and process control: A systematic review," *Computers & Chemical Engineering*, vol. 71, pp. 377 – 390, 2014.

[95] R. Mahadevan, F. J. Doyle, and A. C. Allcock, "Control-relevant scheduling of polymer grade transitions," *AIChE Journal*, vol. 48, pp. 1754–1764, 8 2002.

[96] C. Chatzidoukas, J. Perkins, E. Pistikopoulos, and C. Kiparissides, "Optimal grade transition and selection of closed-loop controllers in a gas-phase olefin polymerization fluidized bed reactor," *Chemical Engineering Science*, vol. 58, no. 16, pp. 3643 – 3658, 2003.

[97] A. Flores-Tlacuahuac and I. E. Grossmann, "Simultaneous cyclic scheduling and control of a multiproduct cstr," *Industrial & Engineering Chemistry Research*, vol. 45, no. 20, pp. 6698–6712, 2006.

[98] S. Terrazas-Moreno, A. Flores-Tlacuahuac, and I. E. Grossmann, "Simultaneous cyclic scheduling and optimal control of polymerization reactors," *AIChE Journal*, vol. 53, pp. 2301–2315, 9 2007.

[99] A. Flores-Tlacuahuac and I. E. Grossmann, "Simultaneous cyclic scheduling and control of tubular reactors: Single production lines," *Industrial & Engineering Chemistry Research*, vol. 49, no. 22, pp. 11453–11463, 2010.

[100] A. Flores-Tlacuahuac and I. E. Grossmann, "Simultaneous cyclic scheduling and control of tubular reactors: Parallel production lines," *Industrial & Engineering Chemistry Research*, vol. 50, no. 13, pp. 8086–8096, 2011.

[101] J. Zhuge and M. G. Ierapetritou, "Integration of scheduling and control with closed loop implementation," *Industrial and Engineering Chemistry Research*, vol. 51, no. 25, pp. 8550–8565, 2012.

[102] M. A. Gutiérrez-Limón, A. Flores-Tlacuahuac, and I. E. Grossmann, "Minlp formulation for simultaneous planning, scheduling, and control of short-period single-unit processing systems," *Industrial & Engineering Chemistry Research*, vol. 53, no. 38, pp. 14679–14694, 2014.

[103] J. Du, J. Park, I. Harjunkoski, and M. Baldea, "A time scale-bridging approach for integrating production scheduling and process control," *Computers & Chemical Engineering*, vol. 79, pp. 59 – 69, 2015.

[104] M. Baldea, J. Du, J. Park, and I. Harjunkoski, "Integrated production scheduling and model predictive control of continuous processes," *AIChE Journal*, vol. 61, pp. 4179–4190, 12 2015.

[105] B. Burnak, J. Katz, N. A. Diangelakis, and E. N. Pistikopoulos, "Simultaneous process scheduling and control: A multiparametric programming-based approach," *Industrial & Engineering Chemistry Research*, vol. 57, no. 11, pp. 3963–3976, 2018.

[106] V. M. Charitopoulos, L. G. Papageorgiou, and V. Dua, "Closed-loop integration of planning, scheduling and multi-parametric nonlinear control," *Computers & Chemical Engineering*, vol. 122, pp. 172 – 192, 2019. 2017 Edition of the European Symposium on Computer Aided Process Engineering (ESCAPE-27).

[107] C. Loeblein and J. D. Perkins, "Structural design for on-line process optimization: I. dynamic economics of mpc," *AIChE Journal*, vol. 45, no. 5, pp. 1018–1029, 1999.

[108] A. C. Zanin, M. T. de Gouvêa, and D. Odloak, "Integrating real-time optimization into the model predictive controller of the fcc system," *Control Engineering Practice*, vol. 10, no. 8, pp. 819 – 831, 2002.

[109] J. B. Rawlings and R. Amrit, *Optimizing Process Economic Performance Using Model Predictive Control*, pp. 119–138. Berlin, Heidelberg: Springer Berlin Heidelberg, 2009.

[110] R. Amrit, J. B. Rawlings, and D. Angeli, "Economic optimization using model predictive control with a terminal cost," *Annual Reviews in Control*, vol. 35, no. 2, pp. 178 – 186, 2011.

[111] L. Würth, R. Hannemann, and W. Marquardt, "A two-layer architecture for economically optimal process control and operation," *Journal of Process Control*, vol. 21, no. 3, pp. 311 – 321, 2011. Thomas McAvoy Festschrift.

[112] M. Ellis and P. D. Christofides, "Selection of control configurations for economic model predictive control systems," *AIChE Journal*, vol. 60, pp. 3230–3242, 9 2014.

[113] M. Z. Jamaludin and C. L. E. Swartz, "Dynamic real-time optimization with closed-loop prediction," *AIChE Journal*, vol. 63, no. 9, pp. 3896–3911, 2017.

[114] H. Li and C. L. E. Swartz, "Dynamic real-time optimization of distributed mpc systems using rigorous closed-loop prediction," *Computers & Chemical Engineering*, vol. 122, pp. 356

– 371, 2019. 2017 Edition of the European Symposium on Computer Aided Process Engineering (ESCAPE-27).

[115] J. M. Simkoff and M. Baldea, "Production scheduling and linear mpc: Complete integration via complementarity conditions," *Computers & Chemical Engineering*, vol. 125, pp. 287 – 305, 2019.

[116] J. Downs and E. Vogel, "A plant-wide industrial process control problem," *Computers & Chemical Engineering*, vol. 17, no. 3, pp. 245 – 255, 1993. Industrial challenge problems in process control.

[117] I. E. Grossmann and R. W. H. Sargent, "Optimum design of heat exchanger networks," *Computers & Chemical Engineering*, vol. 2, no. 1, pp. 1 – 7, 1978.

[118] N. Nishida, G. Stephanopoulos, and A. W. Westerberg, "A review of process synthesis," *AIChE Journal*, vol. 27, pp. 321–351, 5 1981.

[119] K. P. Papalexandri and E. N. Pistikopoulos, "Generalized modular representation framework for process synthesis," *AIChE Journal*, vol. 42, pp. 1010–1032, 4 1996.

[120] S. E. Demirel, J. Li, and M. M. F. Hasan, "Systematic process intensification using building blocks," *Computers & Chemical Engineering*, vol. 105, pp. 2 – 38, 2017. Process Intensification.

[121] A. K. Tula, D. K. Babi, J. Bottlaender, M. R. Eden, and R. Gani, "A computer-aided software-tool for sustainable process synthesis-intensification," *Computers & Chemical Engineering*, vol. 105, pp. 74 – 95, 2017. Process Intensification.

[122] F. E. da Cruz and V. I. Manousiouthakis, "Process intensification of reactive separator networks through the ideas conceptual framework," *Computers & Chemical Engineering*, vol. 105, pp. 39 – 55, 2017. Process Intensification.

[123] Y. Tian and E. N. Pistikopoulos, "Synthesis of operable process intensification systems—steady-state design with safety and operability considerations," *Industrial & Engineering Chemistry Research*, vol. 58, no. 15, pp. 6049–6068, 2019.

[124] S. E. Demirel, J. Li, and M. M. F. Hasan, "Systematic process intensification," *Current Opinion in Chemical Engineering*, 2019.

[125] S. E. Demirel, J. Li, and M. M. F. Hasan, "A general framework for process synthesis, integration, and intensification," *Industrial & Engineering Chemistry Research*, vol. 58, no. 15, pp. 5950–5967, 2019.

[126] Y. Tian, S. E. Demirel, M. M. F. Hasan, and E. N. Pistikopoulos, "An overview of process systems engineering approaches for process intensification: State of the art," *Chemical Engineering and Processing - Process Intensification*, vol. 133, pp. 160 – 210, 2018.

[127] M. Baldea, "From process integration to process intensification," *Computers & Chemical Engineering*, vol. 81, pp. 104 – 114, 2015. Special Issue: Selected papers from the 8th International Symposium on the Foundations of Computer-Aided Process Design (FOCAPD 2014), July 13-17, 2014, Cle Elum, Washington, USA.

[128] Y. Tian and E. N. Pistikopoulos, "Synthesis of operable process intensification systems: advances and challenges," *Current Opinion in Chemical Engineering*, 2019.

[129] L. S. Dias and M. G. Ierapetritou, "Optimal operation and control of intensified processes — challenges and opportunities," *Current Opinion in Chemical Engineering*, 2019.

[130] E. N. Pistikopoulos, N. A. Diangelakis, R. Oberdieck, M. M. Papathanasiou, I. Nascu, and M. Sun, "PAROC – An integrated framework and software platform for the optimisation and advanced model-based control of process systems," *Chemical Engineering Science*, vol. 136, pp. 115–138, 2015.

[131] N. Diangelakis, B. Burnak, and E. Pistikopoulos, "A multi-parametric programming approach for the simultaneous process scheduling and control–application to a domestic cogeneration unit," *Chemical Process Control. Tucson, Arizona, January*, pp. 8–12, 2017.

[132] L. Ricardez-Sandoval, H. Budman, and P. Douglas, "Integration of design and control for chemical processes: A review of the literature and some recent results," *Annual Reviews in Control*, vol. 33, no. 2, pp. 158 – 171, 2009.

[133] M. Soroush and C. Kravaris, "Optimal design and operation of batch reactors. 2. a case study," *Industrial and Engineering Chemistry Research*, vol. 32, no. 5, pp. 882–893, 1993.

[134] M. Soroush and C. Kravaris, "Optimal design and operation of batch reactors. 1. theoretical framework," *Industrial and Engineering Chemistry Research*, vol. 32, no. 5, pp. 866–881, 1993.

[135] K. Papalexandri and E. Pistikopoulos, "Synthesis and retrofit design of operable heat exchanger networks. 1. flexibility and structural controllability aspects," *Industrial and Engineering Chemistry Research*, vol. 33, no. 7, pp. 1718–1737, 1994.

[136] C. Floudas, "Global optimization in design and control of chemical process systems," *Journal of Process Control*, vol. 10, no. 2, pp. 125–134, 2000.

[137] C. Floudas, Z. Gümüş, and M. Ierapetritou, "Global optimization in design under uncertainty: Feasibility test and flexibility index problems," *Industrial and Engineering Chemistry Research*, vol. 40, no. 20, pp. 4267–4282, 2001.

[138] P. Bahri, J. Bandoni, and J. Romagnoli, "Integrated flexibility and controllability analysis in design of chemical processes," *AIChE Journal*, vol. 43, no. 4, pp. 997–1015, 1997.

[139] N. Chatrattanawet, S. Skogestad, and A. Arpornwichanop, "Control structure design and controllability analysis for solid oxide fuel cell," *Chemical Engineering Transactions*, vol. 39, no. Special Issue, pp. 1291–1296, 2014.

[140] J. Gong, G. Hytoft, and R. Gani, "An integrated computer aided system for integrated design and control of chemical processes," *Computers and Chemical Engineering*, vol. 19, no. SUPPL. 1, pp. 489–494, 1995.

[141] P. Vega, R. Lamanna, S. Revollar, and M. Francisco, "Integrated design and control of chemical processes - part ii: An illustrative example," *Computers and Chemical Engineering*, vol. 71, pp. 618–635, 2014.

[142] V. Bansal, R. Ross, J. Perkins, and E. Pistikopoulos, "Interactions of design and control: Double-effect distillation," *Journal of Process Control*, vol. 10, no. 2, pp. 219–227, 2000.

[143] V. Bansal, J. Perkins, and E. Pistikopoulos, "A case study in simultaneous design and control using rigorous, mixed-integer dynamic optimization models," *Industrial and Engineering Chemistry Research*, vol. 41, no. 4, pp. 760–778, 2002.

[144] M. Georgiadis, M. Schenk, E. Pistikopoulos, and R. Gani, "The interactions of design, control and operability in reactive distillation systems," *Computers and Chemical Engineering*, vol. 26, no. 4-5, pp. 735–746, 2002.

[145] V. Bansal, V. Sakizlis, R. Ross, J. D. Perkins, and E. N. Pistikopoulos, "New algorithms for mixed-integer dynamic optimization," *Computers and Chemical Engineering*, vol. 27, no. 5, pp. 647–668, 2003.

[146] V. Sakizlis, J. Perkins, and E. Pistikopoulos, "Recent advances in optimization-based simultaneous process and control design," *Computers and Chemical Engineering*, vol. 28, no. 10, pp. 2069–2086, 2004.

[147] I. Washington and C. Swartz, "Design under uncertainty using parallel multiperiod dynamic optimization," *AIChE Journal*, vol. 60, no. 9, pp. 3151–3168, 2014.

[148] L. A. Ricardez-Sandoval, "Optimal design and control of dynamic systems under uncertainty: A probabilistic approach," *Computers & Chemical Engineering*, vol. 43, pp. 91 – 107, 2012.

[149] A. Flores-Tlacuahuac and L. Biegler, "Simultaneous mixed-integer dynamic optimization for integrated design and control," *Computers and Chemical Engineering*, vol. 31, no. 5-6, pp. 588–600, 2007.

[150] A. Flores-Tlacuahuac and L. Biegler, "Integrated control and process design during optimal polymer grade transition operations," *Computers and Chemical Engineering*, vol. 32, no. 11, pp. 2823–2837, 2008.

[151] I. Kookos and J. Perkins, "An algorithmic method for the selection of multivariable process control structures," *Journal of Process Control*, vol. 12, no. 1, pp. 85–99, 2002.

[152] R.-N. De La Fuente and A. Flores-Tlacuahuac, "Integrated design and control using a simultaneous mixed-integer dynamic optimization approach," *Industrial and Engineering Chemistry Research*, vol. 48, no. 4, pp. 1933–1943, 2009.

[153] Y. Chen, T. Adams, and P. Barton, "Optimal design and operation of flexible energy polygeneration systems," *Industrial and Engineering Chemistry Research*, vol. 50, no. 8, pp. 4553–4566, 2011.

[154] Y. Chen, T. Adams, and P. Barton, "Optimal design and operation of static energy polygeneration systems," *Industrial and Engineering Chemistry Research*, vol. 50, no. 9, pp. 5099–5113, 2011.

[155] X. Li and P. Barton, "Optimal design and operation of energy systems under uncertainty," *Journal of Process Control*, vol. 30, pp. 1–9, 2015.

[156] L. Zhang, C. Xue, A. Malcolm, K. Kulkarni, and A. Linninger, "Distributed system design under uncertainty," *Industrial and Engineering Chemistry Research*, vol. 45, no. 25, pp. 8352–8360, 2006.

[157] P. Vega, R. Lamanna de Rocco, S. Revollar, and M. Francisco, "Integrated design and control of chemical processes - part i: Revision and classification," *Computers and Chemical Engineering*, vol. 71, pp. 602–617, 2014.

[158] L. Ricardez-Sandoval, "Current challenges in the design and control of multiscale systems," *Canadian Journal of Chemical Engineering*, vol. 89, no. 6, pp. 1324–1341, 2011.

[159] Z. Yuan, B. Chen, G. Sin, and R. Gani, "State-of-the-art and progress in the optimization-based simultaneous design and control for chemical processes," *AIChE Journal*, vol. 58, no. 6, pp. 1640–1659, 2012.

[160] A. Ghobeity and A. Mitsos, "Optimal design and operation of desalination systems: New challenges and recent advances," *Current Opinion in Chemical Engineering*, vol. 6, pp. 61–68, 2014.

[161] R. Oberdieck, N. A. Diangelakis, M. M. Papathanasiou, I. Nascu, and E. N. Pistikopoulos, "POP - Parametric Optimization Toolbox," *Industrial and Engineering Chemistry Research*, vol. 55, no. 33, pp. 8979–8991, 2016.

[162] R. Oberdieck, N. A. Diangelakis, S. Avraamidou, and E. N. Pistikopoulos, "On unbounded and binary parameters in multi-parametric programming: applications to mixed-integer bilevel optimization and duality theory," *Journal of Global Optimization*, 2016, in press.

[163] V. M. Charitopoulos and V. Dua, "Explicit model predictive control of hybrid systems and multiparametric mixed integer polynomial programming," *AIChE Journal*, vol. 62, no. 9, pp. 3441–3460, 2016.

[164] S. Engell and I. Harjunkoski, "Optimal operation: Scheduling, advanced control and their integration," *Computers and Chemical Engineering*, vol. 47, pp. 121–133, 2012.

[165] I. Harjunkoski, R. Nyström, and A. Horch, "Integration of scheduling and control-Theory or practice?," *Computers and Chemical Engineering*, vol. 33, no. 12, pp. 1909–1918, 2009.

[166] M. A. Gutiérrez-Limón, A. Flores-Tlacuahuac, and I. E. Grossmann, "A Multiobjective Optimization Approach for the Simultaneous Single Line Scheduling and Control of CSTRs," *Industrial & Engineering Chemistry Research*, vol. 51, no. 17, pp. 5881–5890, 2012.

[167] M. A. Gutiérrez-Limón, A. Flores-Tlacuahuac, and I. E. Grossmann, "MINLP formulation for simultaneous planning, scheduling, and control of short-period single-unit processing systems," *Industrial and Engineering Chemistry Research*, vol. 53, no. 38, pp. 14679–14694, 2014.

[168] K. Mitra, R. D. Gudi, S. C. Patwardhan, and G. Sardar, "Resiliency issues in integration of scheduling and control," *Industrial and Engineering Chemistry Research*, vol. 49, no. 1, pp. 222–235, 2010.

[169] Y. Nie, L. T. Biegler, and J. M. Wassick, "Integrated Scheduling and Dynamic Optimization of Batch Processes Using State Equipment Networks," *AIChE Journal*, vol. 58, no. 11, pp. 3416–3432, 2012.

[170] Y. Nie, L. T. Biegler, C. M. Villa, and J. M. Wassick, "Discrete time formulation for the integration of scheduling and dynamic optimization," *Industrial & Engineering Chemistry Research*, vol. 54, no. 16, pp. 4303–4315, 2015.

[171] E. Capón-García, G. Guillén-Gosálbez, and A. Espuña, "Integrating process dynamics within batch process scheduling via mixed-integer dynamic optimization," *Chemical Engineering Science*, vol. 102, pp. 139–150, 2013.

[172] Y. Chu and F. You, "Integration of production scheduling and dynamic optimization for multi-product CSTRs: Generalized Benders decomposition coupled with global mixed-integer fractional programming," *Computers and Chemical Engineering*, vol. 58, pp. 315–333, 2013.

[173] R. C. Pattison, C. R. Touretzky, T. Johansson, I. Harjunkoski, and M. Baldea, "Optimal Process Operations in Fast-Changing Electricity Markets: Framework for Scheduling with Low-Order Dynamic Models and an Air Separation Application," *Industrial and Engineering Chemistry Research*, vol. 55, no. 16, pp. 4562–4584, 2016.

[174] M. T. Kelley, R. C. Pattison, R. Baldick, and M. Baldea, "An efficient milp framework for integrating nonlinear process dynamics and control in optimal production scheduling calculations," *Computers & Chemical Engineering*, vol. 110, pp. 35 – 52, 2018.

[175] Y. Chu and F. You, "Integration of scheduling and control with online closed-loop implementation: Fast computational strategy and large-scale global optimization algorithm," *Computers and Chemical Engineering*, vol. 47, pp. 248–268, 2012.

[176] J. G. Costandy, T. F. Edgar, and M. Baldea, "A scheduling perspective on the monetary value of improving process control," *Computers and Chemical Engineering*, vol. 112, pp. 121–131, 2018.

[177] J. Zhuge and M. G. Ierapetritou, "Integration of scheduling and control for batch processes using multi-parametric model predictive control," *AIChE Journal*, vol. 60, no. 9, pp. 3169–3183, 2014.

[178] L. S. Dias, R. C. Pattison, C. Tsay, M. Baldea, and M. G. Ierapetritou, "A simulation-based optimization framework for integrating scheduling and model predictive control, and its application to air separation units," *Computers & Chemical Engineering*, vol. 113, pp. 139 – 151, 2018.

[179] M. Ellis and P. D. Christofides, "Integrating dynamic economic optimization and model predictive control for optimal operation of nonlinear process systems," *Control Engineering Practice*, vol. 22, pp. 242 – 251, 2014.

[180] M. Ellis and P. D. Christofides, "Real-time economic model predictive control of nonlinear process systems," *AIChE Journal*, vol. 61, no. 2, pp. 555–571, 2015.

[181] A. Alanqar, H. Durand, F. Albalawi, and P. D. Christofides, "An economic model predictive control approach to integrated production management and process operation," *AIChE Journal*, vol. 63, no. 6, pp. 1892–1906, 2017.

[182] L. D. Beal, D. Petersen, D. Grimsman, S. Warnick, and J. D. Hedengren, "Integrated scheduling and control in discrete-time with dynamic parameters and constraints," *Computers & Chemical Engineering*, vol. 115, pp. 361 – 376, 2018.

[183] A. Huercio, A. Espuña, and L. Puigjaner, "Incorporating on-line scheduling strategies in integrated batch production control," *Computers and Chemical Engineering*, vol. 19, no. SUPPL. 1, pp. 609–614, 1995.

[184] K. Subramanian, C. T. Maravelias, and J. B. Rawlings, "A state-space model for chemical production scheduling," *Computers & Chemical Engineering*, vol. 47, pp. 97–110, 2012.

[185] K. Subramanian, J. B. Rawlings, C. T. Maravelias, J. Flores-Cerrillo, and L. Megan, "Integration of control theory and scheduling methods for supply chain management," *Computers and Chemical Engineering*, vol. 51, pp. 4–20, 2013.

[186] G. M. Kopanos, M. C. Georgiadis, and E. N. Pistikopoulos, "Energy production planning of a network of micro combined heat and power generators," *Applied Energy*, vol. 102, pp. 1522–1534, 2013.

[187] G. M. Kopanos and E. N. Pistikopoulos, "Reactive scheduling by a multiparametric programming rolling horizon framework: A case of a network of combined heat and power units," *Industrial and Engineering Chemistry Research*, vol. 53, no. 11, pp. 4366–4386, 2014.

[188] C. R. Touretzky and M. Baldea, "Integrating scheduling and control for economic mpc of buildings with energy storage," *Journal of Process Control*, vol. 24, no. 8, pp. 1292 – 1300, 2014. Economic nonlinear model predictive control.

[189] S. Liu and J. Liu, "Economic model predictive control with extended horizon," *Automatica*, vol. 73, pp. 180 – 192, 2016.

[190] M. Bassett, P. Dave, F. Doyle, G. Kudva, J. Pekny, G. Reklaitis, S. Subrahmanyam, D. Miller, and M. Zentner, "Perspectives on model based integration of process operations," *Computers & Chemical Engineering*, vol. 20, no. 6, pp. 821–844, 1996.

[191] M. Ellis, H. Durand, and P. D. Christofides, "A tutorial review of economic model predictive control methods," *Journal of Process Control*, vol. 24, no. 8, pp. 1156 – 1178, 2014. Economic nonlinear model predictive control.

[192] Y. Chu and F. You, "Model-based integration of control and operations: Overview, challenges, advances, and opportunities," *Computers and Chemical Engineering*, vol. 83, pp. 2–20, 2015.

[193] L. S. Dias and M. G. Ierapetritou, "Integration of scheduling and control under uncertainties: Review and challenges," *Chemical Engineering Research and Design*, vol. 116, pp. 98–113, 2016.

[194] L. S. Dias and M. G. Ierapetritou, "From process control to supply chain management: An overview of integrated decision making strategies," *Computers & Chemical Engineering*, vol. 106, pp. 826 – 835, 2017.

[195] J. Löfberg, "Yalmip: A toolbox for modeling and optimization in matlab," *Proceedings of the IEEE International Symposium on Computer-Aided Control System Design*, pp. 284–289, 2004.

[196] P. Grieder, M. Kvasnica, M. Baotic, and M. Morari, "Low Complexity Control of Piecewise Affine Systems with Stability Guarantee," *Proceedings of the 2004 American Control Conference, Boston, Massachusetts*, pp. 1196–1201, 2004.

[197] G. Towler and R. Sinnott, "Chapter 7 - capital cost estimating," in *Chemical Engineering Design (Second Edition)* (G. Towler and R. Sinnott, eds.), pp. 307 – 354, Boston: Butterworth-Heinemann, second edition ed., 2013.

[198] N. Diangelakis, C. Panos, and E. Pistikopoulos, "Design optimization of an internal combustion engine powered chp system for residential scale application," *Computational Management Science*, vol. 11, no. 3, pp. 237–266, 2014.

[199] N. A. Diangelakis, S. Avraamidou, and E. N. Pistikopoulos, "Decentralized Multiparametric Model Predictive Control for Domestic Combined Heat and Power Systems," *Industrial & Engineering Chemistry Research*, vol. 55, no. 12, pp. 3313–3326, 2016.

[200] J. Katz, B. Burnak, and E. N. Pistikopoulos, "The impact of model approximation in multiparametric model predictive control," *Chemical Engineering Research and Design*, vol. 139, pp. 211 – 223, 2018.

[201] C. T. Maravelias, "General framework and modeling approach classification for chemical production scheduling," *AIChE Journal*, vol. 58, no. 6, pp. 1812–1828, 2012.

[202] M. G. Ierapetritou and C. A. Floudas, "Effective continuous-time formulation for short-term scheduling. 1. multipurpose batch processes," *Industrial & Engineering Chemistry Research*, vol. 37, no. 11, pp. 4341–4359, 1998.

[203] C. C. Pantelides, "Unified frameworks for optimal process planning and scheduling," in *Proceedings on the second conference on foundations of computer aided operations*, pp. 253–274, Cache Publications New York, 1994.

[204] C. A. Floudas and X. Lin, "Continuous-time versus discrete-time approaches for scheduling of chemical processes: A review," *Computers and Chemical Engineering*, vol. 28, no. 11, pp. 2109–2129, 2004.

[205] C. T. Maravelias and I. E. Grossmann, "New general continuous-time state-task network formulation for short-term scheduling of multipurpose batch plants," *Industrial & Engineering Chemistry Research*, vol. 42, no. 13, pp. 3056–3074, 2003.

[206] R. Seid and T. Majozi, "A robust mathematical formulation for multipurpose batch plants," *Chemical Engineering Science*, vol. 68, no. 1, pp. 36 – 53, 2012.

[207] A. F. Merchan and C. T. Maravelias, "Reformulations of mixed-integer programming continuous-time models for chemical production scheduling," *Industrial & Engineering Chemistry Research*, vol. 53, no. 24, pp. 10155–10165, 2014.

[208] H. Lee and C. T. Maravelias, "Combining the advantages of discrete- and continuous-time scheduling models: Part 1. framework and mathematical formulations," *Computers & Chemical Engineering*, vol. 116, pp. 176 – 190, 2018.

[209] H. Mostafaei and I. Harjunkoski, "Continuous-time scheduling formulation for multipurpose batch plants," *AIChE Journal*, vol. 66, no. 2, 2020.

[210] M. Rafiei and L. A. Ricardez-Sandoval, "New frontiers, challenges, and opportunities in integration of design and control for enterprise-wide sustainability," *Computers & Chemical Engineering*, vol. 132, p. 106610, 2020.

[211] Y. Chu and F. You, "Integrated planning, scheduling, and dynamic optimization for batch processes: Minlp model formulation and efficient solution methods via surrogate modeling," *Industrial & Engineering Chemistry Research*, vol. 53, no. 34, pp. 13391–13411, 2014.

[212] Y. I. Valdez-Navarro and L. A. Ricardez-Sandoval, "A novel back-off algorithm for integration of scheduling and control of batch processes under uncertainty," *Industrial & Engineering Chemistry Research*, vol. 58, no. 48, pp. 22064–22083, 2019.

[213] F. Rossi, D. Casas-Orozco, G. Reklaitis, F. Manenti, and G. Buzzi-Ferraris, "A computational framework for integrating campaign scheduling, dynamic optimization and optimal control in multi-unit batch processes," *Computers & Chemical Engineering*, vol. 107, pp. 184 – 220, 2017. In honor of Professor Rafiqul Gani.

[214] D. Rippin, "Design and operation of multiproduct and multipurpose batch chemical plants. — an analysis of problem structure," *Computers & Chemical Engineering*, vol. 7, no. 4, pp. 463 – 481, 1983.

[215] Y. Chu and F. You, "Integration of scheduling and dynamic optimization of batch processes under uncertainty: Two-stage stochastic programming approach and enhanced generalized benders decomposition algorithm," *Industrial & Engineering Chemistry Research*, vol. 52, no. 47, pp. 16851–16869, 2013.

[216] M. S. Díaz and L. T. Biegler, "Chapter 17 - dynamic optimization in process systems," in *Introduction to Software for Chemical Engineers* (M. M. Martín, ed.), pp. 681 – 711, CRC Press/Taylor & Francis, second ed., 2019.

[217] A. Wächter and L. T. Biegler, "On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming," *Mathematical Programming*, vol. 106, pp. 25–57, Mar 2006.

[218] W. E. Hart, J.-P. Watson, and D. L. Woodruff, "Pyomo: modeling and solving mathematical programs in python," *Mathematical Programming Computation*, vol. 3, no. 3, pp. 219–260, 2011.

[219] W. E. Hart, C. D. Laird, J.-P. Watson, D. L. Woodruff, G. A. Hackebeil, B. L. Nicholson, and J. D. Siirola, *Pyomo–optimization modeling in python*, vol. 67. Springer Science & Business Media, second ed., 2017.

[220] B. Nicholson, J. D. Siirola, J.-P. Watson, V. M. Zavala, and L. T. Biegler, "pyomo.dae: a modeling and automatic discretization framework for optimization with differential and

algebraic equations," *Mathematical Programming Computation*, vol. 10, no. 2, pp. 187–223, 2018.

[221] M. Tawarmalani and N. V. Sahinidis, "A polyhedral branch-and-cut approach to global optimization," *Mathematical Programming*, vol. 103, pp. 225–249, 2005.

[222] B. Burnak, J. Katz, N. A. Diangelakis, and E. N. Pistikopoulos, "Integration of design, scheduling, and control of combined heat and power systems: A multiparametric programming based approach," in *Computer Aided Chemical Engineering*, vol. 44, pp. 2203–2208, Elsevier, 2018.

[223] E. N. Pistikopoulos, "Perspectives in multiparametric programming and explicit model predictive control," *AIChE Journal*, vol. 55, no. 8, pp. 1918–1925, 2009.

[224] M. Wittmann-Hohlbein and E. N. Pistikopoulos, "Approximate solution of mp-milp problems using piecewise affine relaxation of bilinear terms," *Computers & Chemical Engineering*, vol. 61, pp. 136 – 155, 2014.

[225] A. Alessio and A. Bemporad, "A survey on explicit model predictive control," *Lecture Notes in Control and Information Sciences*, vol. 384, pp. 345–369, 2009.

[226] M. M. Papathanasiou, A. L. Quiroga-Campano, F. Steinebach, M. Elviro, A. Mantalaris, and E. N. Pistikopoulos, "Advanced model-based control strategies for the intensification of upstream and downstream processing in mab production," *Biotechnology Progress*, vol. 33, no. 4, pp. 966–988, 2017.

[227] J. Lee and H. J. Chang, "Explicit model predictive control for linear time-variant systems with application to double-lane-change maneuver," *PLoS ONE*, vol. 13, no. 12, 2018.

[228] A. Shokry, C. Dombayci, and A. Espuña, "Multiparametric metamodels for model predictive control of chemical processes," *Computer Aided Chemical Engineering*, vol. 38, pp. 937–942, 2016.

[229] M. Onel, B. Burnak, and E. N. Pistikopoulos, "Integrated data-driven process monitoring and explicit fault-tolerant multiparametric control," *Industrial & Engineering Chemistry Research*, vol. 59, no. 6, pp. 2291–2306, 2020.

[230] M. Wittmann-Hohlbein and E. N. Pistikopoulos, "Proactive scheduling of batch processes by a combined robust optimization and multiparametric programming approach," *AIChE Journal*, vol. 59, no. 11, pp. 4184–4211, 2013.

[231] Y. Tian, I. Pappas, B. Burnak, J. Katz, and E. N. Pistikopoulos, "A systematic framework for the synthesis of operable process intensification systems – reactive separation systems," *Computers & Chemical Engineering*, vol. 134, p. 106675, 2020.

[232] M. Köppe, M. Queyranne, and C. T. Ryan, "Parametric integer programming algorithm for bilevel mixed integer programs," *Journal of Optimization Theory and Applications*, vol. 146, pp. 137–150, Jul 2010.

[233] S. Avraamidou and E. N. Pistikopoulos, "Multi-parametric global optimization approach for tri-level mixed-integer linear optimization problems," *Journal of Global Optimization*, May 2018.

[234] E. C. Mid and V. Dua, "Parameter estimation using multiparametric programming for implicit euler's method based discretization," *Chemical Engineering Research and Design*, pp. 62–77, 2019.

[235] T. Gal and J. Nedoma, "Multiparametric linear programming," *Management Science*, vol. 18, no. 7, pp. 406–422, 1972.

[236] P. Tøndel, T. A. Johansen, and A. Bemporad, "An algorithm for multi-parametric quadratic programming and explicit mpc solutions," *Automatica*, vol. 39, no. 3, pp. 489–497, 2003.

[237] J. Spjøtvold, E. C. Kerrigan, C. N. Jones, P. Tøndel, and T. A. Johansen, "On the facet-to-facet property of solutions to convex parametric quadratic programs," *Automatica*, vol. 42, no. 12, pp. 2209–2214, 2006.

[238] A. Gupta, S. Bhartiya, and P. S. V. Nataraj, "A novel approach to multiparametric quadratic programming," *Automatica*, vol. 47, no. 9, pp. 2112 – 2117, 2011.

[239] P. Ahmadi-Moshkenani, T. A. Johansen, and S. Olaru, "Combinatorial approach toward multiparametric quadratic programming based on characterizing adjacent critical regions," *IEEE Transactions on Automatic Control*, vol. 63, no. 10, pp. 3221–3231, 2018.

[240] R. Oberdieck, N. A. Diangelakis, and E. N. Pistikopoulos, "Explicit model predictive control: A connected-graph approach," *Automatica*, vol. 76, pp. 103–112, 02 2017.

[241] R. Oberdieck, M. Wittmann-Hohlbein, and E. N. Pistikopoulos, "A branch and bound method for the solution of multiparametric mixed integer linear programming problems," *Journal of Global Optimization*, vol. 59, no. 2-3, pp. 527–543, 2014.

[242] J. D. na, M. K. co, F. J. cek, and M. Kvasnica, "Optimal control of a laboratory binary distillation column via regionless explicit mpc," *Computers & Chemical Engineering*, vol. 96, pp. 139 – 148, 2017.

[243] A. V. Fiacco, "Chapter 2 – basic sensitivity and stability results," in *Introduction to Sensitivity and Stability Analysis in Nonlinear Programming*, vol. 165 of *Mathematics in Science and Engineering*, pp. 8 – 64, Elsevier, 1983.

[244] V. Dua, N. A. Bozinis, and E. N. Pistikopoulos, "A multiparametric programming approach for mixed-integer quadratic engineering problems," *Computers & Chemical Engineering*, vol. 26, no. 4, pp. 715 – 733, 2002.

[245] T. Gal, *Postoptimal Analyses, Parametric Programming, and Related Topics: Degeneracy, Multicriteria Decision Making, Redundancy*. Walter de Gruyter, Berlin, 1995.

[246] M. de Berg, O. Cheong, M. van Kreveld, and M. Overmars, "Delaunay triangulations," in *Computational Geometry: Algorithms and Applications*, pp. 191–218, Berlin, Heidelberg: Springer Berlin Heidelberg, 2008.

[247] B. Grünbaum, *Convex Polytopes*. New York, NY, USA: Springer, 2003.

[248] C. A. Floudas, *Nonlinear and Mixed-Integer Optimization: Fundamentals and Applications*. Topics in Chemical Engineering, Oxford University Press, 1995.

[249] S. Boyd and L. Vandenberghe, *Convex Optimization*. New York, NY, USA: Cambridge University Press, 2004.

[250] A. Akbari and P. I. Barton, "An improved multi-parametric programming algorithm for flux balance analysis of metabolic networks," *Journal of Optimization Theory and Applications*, vol. 178, pp. 502–537, 8 2018.

[251] F. Borrelli, A. Bemporad, and M. Morari, "Geometric algorithm for multiparametric linear programming," *Journal of Optimization Theory and Applications*, vol. 118, pp. 515–540, Sep 2003.

[252] P. Dua and M. C. Georgiadis, *Multiparametric Mixed-Integer Linear Programming*, ch. 3, pp. 53–71. John Wiley & Sons, Ltd, 2011.

[253] R. Oberdieck and E. N. Pistikopoulos, "Multi-objective optimization with convex quadratic cost functions: A multi-parametric programming approach," *Computers and Chemical Engineering*, vol. 85, pp. 36–39, 2016.

[254] M. Herceg, M. Kvasnica, C. Jones, and M. Morari, "Multi-parametric toolbox 3.0," in *Proc. of the European Control Conference*, (Zürich, Switzerland), pp. 502–510, July 17–19 2013.

[255] N. A. Diangelakis, I. S. Pappas, and E. N. Pistikopoulos, "On multiparametric/explicit nmpc for quadratically constrained problems," *IFAC-PapersOnLine*, vol. 51, no. 20, pp. 400 – 405, 2018. 6th IFAC Conference on Nonlinear Model Predictive Control NMPC 2018.

[256] "Economic indicators," *Chemical Engineering*, vol. 125, p. 64, 7 2018.

[257] M. Onel, C. A. Kieslich, Y. A. Guzman, C. A. Floudas, and E. N. Pistikopoulos, "Big data approach to batch process monitoring:simultaneous fault detection and diagnosis using nonlinear support vector machine-based feature selection," *Computers & chemical engineering*, 2018.

[258] M. Onel, C. A. Kieslich, Y. A. Guzman, and E. N. Pistikopoulos, "Simultaneous fault detection and identification in continuous processes via nonlinear support vector machine based feature selection," in *13th International Symposium on Process Systems Engineering (PSE 2018)* (M. R. Eden, M. G. Ierapetritou, and G. P. Towler, eds.), vol. 44 of *Computer Aided Chemical Engineering*, pp. 2077 – 2082, Elsevier, 2018.

[259] E. N. Pistikopoulos, M. C. Georgiadis, and V. Dua, *Multi-parametric programming*, vol. 1. 2011.

[260] G. R. Marseglia and D. M. Raimondo, "Active fault diagnosis: A multi-parametric approach," *Automatica*, vol. 79, pp. 223–230, 2017.

[261] T. Schneider, G. Ducard, K. Rudin, and P. Strupler, "Fault-tolerant control allocation for multirotor helicopters using parametric programming," *rN*, vol. 1, p. r2, 2012.

[262] J. Spjøtvold and T. A. Johansen, "Fault tolerant control allocation for a thruster-controlled floating platform using parametric programming," in *Proceedings of the 48h IEEE Conference on Decision and Control (CDC) held jointly with 2009 28th Chinese Control Conference*, pp. 3311–3317, IEEE, 2009.

[263] Y. Guzman, *Theoretical Advances in Robust Optimization, Feature Selection, and Biomarker Discovery*. PhD thesis, Princeton University, Princeton, NJ, 2016.

[264] M. Onel, C. A. Kieslich, and E. N. Pistikopoulos, "A nonlinear support vector machine-based feature selection approach for fault detection and diagnosis: Application to the tennessee eastman process," *AIChE Journal*, vol. 65, no. 3, pp. 992–1005, 2019.

[265] G. Birol, C. Ündey, and A. Çinar, "A modular simulation package for fed-batch fermentation: Penicillin production," *Computers & Chemical Engineering*, vol. 26, no. 11, pp. 1553–1565, 2002.

[266] J. Van Impe and G. Gins, "An extensive reference dataset for fault detection and identification in batch processes," *Chemometrics and Intelligent Laboratory Systems*, vol. 148, pp. 20–31, 2015.

[267] G. Gins, J. Vanlaer, P. Van den Kerkhof, and J. F. Van Impe, "The raymond simulation package—generating raypresentative monitoring data to design advanced process monitoring and control algorithms," *Computers & Chemical Engineering*, vol. 69, pp. 108–118, 2014.

[268] L. Breiman, "Random forests," *Machine learning*, vol. 45, no. 1, pp. 5–32, 2001.

[269] P. Nomikos and J. F. MacGregor, "Monitoring batch processes using multiway principal component analysis," *AIChE Journal*, vol. 40, no. 8, pp. 1361–1375, 1994.

APPENDIX A

SUPPORTING INFORMATION

A.1 Surrogate models from the single CSTR example presented in Section 3.3

The identified closed loop state space models are as follows.

Surrogate model 1

$$x_{t+1} = \begin{bmatrix} 0.004 & -0.001 & 0.002 \\ -0.031 & -0.010 & 0.045 \\ -0.118 & -0.026 & 0.118 \end{bmatrix} x_t + \begin{bmatrix} -7.2 \\ -4.7 \\ -3.1 \end{bmatrix} 10^{-4} Q_{total,t}$$
$$y_t = \begin{bmatrix} 0.340 & -0.037 & 0.066 \\ 0.072 & -0.040 & 0.031 \\ 0.048 & -0.042 & 0.041 \end{bmatrix} x_t \tag{A.1}$$

Surrogate model 2

$$x_{t+1} = \begin{bmatrix} 0.045 & 0.027 & -0.012 \\ 0.089 & -0.022 & -0.035 \\ 0.027 & 0.021 & -0.092 \end{bmatrix} x_t + \begin{bmatrix} 2.4 \cdot 10^{-4} & 0.130 \\ 9.0 \cdot 10^{-5} & -0.749 \\ 2.9 \cdot 10^{-6} & -0.716 \end{bmatrix} \begin{bmatrix} Q_{total,t} \\ C^{SP}_{P_2,t} \end{bmatrix}$$
$$y_t = \begin{bmatrix} 0.105 & -0.038 & -0.018 \\ 0.738 & -1.005 & -0.381 \\ 0 & 0 & 0 \end{bmatrix} x_t \tag{A.2}$$

Surrogate model 3

$$x_{t+1} = \begin{bmatrix} -0.011 & -0.012 & -0.016 \\ -0.067 & 0.112 & 0.117 \\ 0.134 & -0.148 & 0.220 \end{bmatrix} x_t + \begin{bmatrix} 2.5 \cdot 10^{-4} & 0.171 \\ 9.8 \cdot 10^{-5} & -0.620 \\ -5.5 \cdot 10^{-5} & 0.192 \end{bmatrix} \begin{bmatrix} Q_{total,t} \\ C^{SP}_{P_3,t} \end{bmatrix}$$
$$y_t = \begin{bmatrix} 0.014 & -0.008 & 0.004 \\ 0 & 0 & 0 \\ 0.516 & -1.081 & 0.477 \end{bmatrix} x_t \tag{A.3}$$

where x_t is the identified states, and y_t is the product concentrations $(C_{P_j,t})$.

A.2 Step by step description of the framework

The following are the detailed steps of the PAROC framework tailored to address the integrated design, scheduling, and control problem, describing the derivation of (i) design dependent, schedule-aware controller, (ii) design dependent, control-aware schedule, and (iii) optimal design based on the offline control and scheduling policies, all summarized in Figure A.1. The interplay between the offline decision layers and the information flow in the overall MIDO formulation is illustrated in Figure A.1a. The derivation of the explicit MPC is explained schematically in Figure A.1b. Lastly, the derivation of the offline scheduler is summarized in Figure A.1c. The steps are detailed through the single CSTR example in Chapter 4.

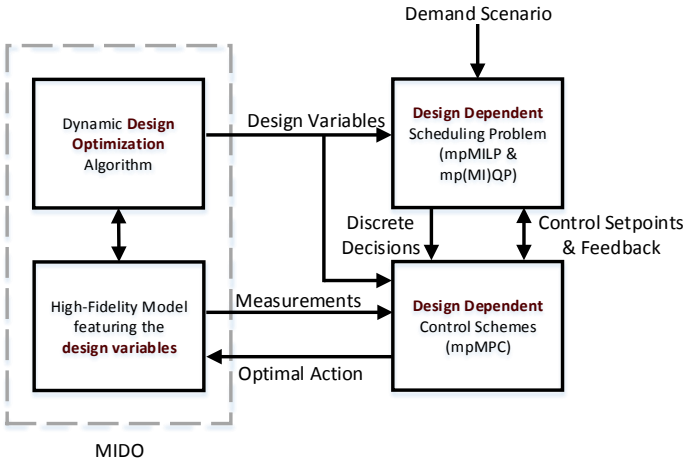
A.2.1 Design dependent, schedule-aware controller

Dynamic high fidelity model. The dynamic equations for the multiproduct CSTR are given as follows. First, we describe the mole balances for the reactants and products and the power law kinetics for the elementary reactions.

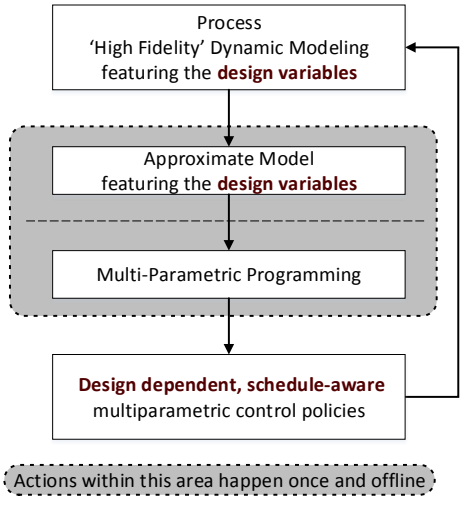
$$\begin{aligned} \frac{dC_{R_i}}{dt} &= \frac{Q_{R_i}C^f_{R_i} - Q_{total}C_{R_i}}{V} + \mathcal{R}_{R_i} \\ \frac{dC_{P_j}}{dt} &= \frac{Q_{total}(C_{P_j} - C_{P_j})}{V} + \mathcal{R}_{P_j} \end{aligned} \tag{A.4}$$

$$\begin{aligned} \mathcal{R}_{R_1} &= -2\mathcal{R}_{P_1} - \mathcal{R}_{P_2} - \mathcal{R}_{P_3} \\ \mathcal{R}_{R_2} &= -\mathcal{R}_{P_2} \\ \mathcal{R}_{R_3} &= -\mathcal{R}_{P_3} \\ \mathcal{R}_{P_1} &= k_1 C^2_{R_1} \\ \mathcal{R}_{P_2} &= k_2 C_{R_1} C_{R_2} \\ \mathcal{R}_{P_3} &= k_3 C_{R_1} C_{R_3} \end{aligned} \tag{A.5}$$

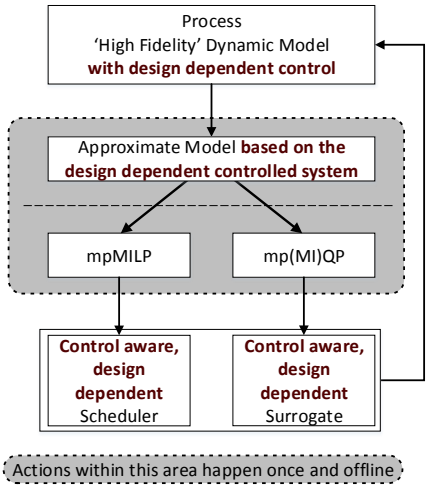
where C denotes the concentration, Q is the volumetric flow rate, V is the volume of the CSTR, $\mathcal{R}$ is the reaction rate, superscript f denotes the feed to the CSTR, R_i and P_j are the indices for the i^{th} reactant and j^{th} product, respectively. The system parameters are given in Table A.1.



(a) Interplay between the process design, scheduling, and control layers.



(b) Explicit model predictive control as a function of design and scheduling actions. The offline strategy is verified against the high fidelity model.



(c) Explicit scheduling as a function of design decisions and aware of closed loop dynamics. The offline strategy is verified against the high fidelity model.

Figure A.1: The integration of process design, scheduling, and control decisions via multiparametric optimization (Reprinted with permission from [8]).

Table A.1: Parameters of the high-fidelity CSTR model (Reprinted with permission from [8]).

Reaction rate constants	Value	Reactant concentration at the feed	Value
k_1	0.1	$C^f_{R_1}$	1.0
k_2	0.9	$C^f_{R_2}$	0.8
k_3	1.5	$C^f_{R_3}$	1.0

The total volumetric flow rate is defined as the sum of reactant flow rates at the inlet of the reactor. Note that constant volume reactor is assumed, therefore the total flow rate at the inlet is equal to the total flow rate at the outlet.

$$Q_{total} = \sum_i Q_{R_i} \tag{A.6}$$

The inventory levels of the product of interest is as follows.

$$\frac{dW_{P_j}}{dt} = \begin{cases} Q_{total} C_{P_j} - DR_{P_j}, \text{ if } Pur_{P_j} \geq 0.90 \\ -DR_{P_j}, \text{ if } Pur_{P_j} < 0.90 \end{cases} \tag{A.7}$$

where W_{P_j} is the inventory level, DR_{P_j} is the demand rate, and Pur_{P_j} is the purity level in the CSTR as defined in Equation A.8.

$$Pur_{P_j} = \frac{C_{P_j}}{\sum_j C_{P_j}} \tag{A.8}$$

The molar fractions of the reactant flow rates are defined in Equation A.9. Note that the molar fractions are utilized as the manipulated variables in the mpMPC control scheme, as demonstrated in the following sections.

$$\begin{aligned} a_{R_i} &= \frac{Q_{R_i}}{Q_{total}} \\ \sum_i a_{R_i} &= 1 \end{aligned} \tag{A.9}$$

Model approximation. The input space is partitioned into multiple mutually exclusive subspaces which are concatenated into a single state space representation. Note that using a single state space model model maintains the identical design dependence in every constituent subspace. The input space is set up as presented in Eq. A.10.

$$
\begin{aligned}
u &= \left[u_1, u_2, u_3, u_4\right]^T \\
u_1 &= a_2,\ a_2 \in [0, 0.5) \\
u_2 &= a_2,\ a_2 \in [0.5, 1] \\
u_3 &= a_3,\ a_3 \in [0, 0.55) \\
u_4 &= a_3,\ a_3 \in [0.55, 1]
\end{aligned}
\tag{A.10}
$$

where a_2 and a_3 are the volumetric fractions of R_2 and R_3 in the feed stream, respectively. MATLAB® System Identification Toolbox™is used to derive the state space matrices model as follows.

$$
A \cdot 10^4 = \begin{bmatrix}
9670 & 8.00 & -23.0 & -915 & -23.1 & -20.0 & 5.12 \\
21.0 & 9650 & -119 & 83.6 & -793 & 47.5 & -389 \\
74.9 & -75.3 & 8610 & 214 & -1060 & 347 & 1360 \\
656 & 172 & 127 & 9540 & 104 & 2270 & -309 \\
187 & 179 & 219 & 3.88 & 9850 & 879 & -398 \\
-1440 & -151 & -193 & -741 & -138 & 5650 & 148 \\
149 & 1130 & -36.5 & -31.8 & -82.0 & 57.9 & 6950
\end{bmatrix}
\tag{A.11a}
$$

$$B \cdot 10^3 = \begin{bmatrix} 2.35 & -1.47 & -1.82 & 3.36 \\ 1.46 & -1.36 & 3.38 & -1.76 \\ 2.69 & -5.12 & 7.85 & -7.13 \\ -7.75 & 5.44 & 16.2 & 6.43 \\ -3.56 & 2.36 & 5.31 & 1.37 \\ 15.1 & -13.6 & -34.1 & -5.63 \\ -12.8 & -40.8 & -11.0 & -3.00 \end{bmatrix} \tag{A.11b}$$

$$C \cdot 10^8 = \begin{bmatrix} -537 & 95.4 \\ 532 & -105 \\ -5070 & 944 \\ -2500 & 415 \\ -89.6 & 6.44 \\ 4860 & -822 \\ 5890 & -1070 \end{bmatrix} \tag{A.11c}$$

$$D \cdot 10^2 = \begin{bmatrix} 2.73 & -748 & 303 & 202 & -816 & 613 & 49.4 \\ 582 & 497 & 90.8 & -31.6 & -30.6 & 1.61 & 4.02 \\ -705 & 730 & 134 & 28.2 & -43.3 & 5.46 & 4.80 \end{bmatrix} \tag{A.11d}$$

Design of the mpMPC. The control scheme is based on the standard MPC formulation given by Eq. A.12, which features two major additions for this specific application, namely (i) incorporation of mutually exclusive control decisions, (ii) introduction of soft constraints to minimize the transition time in the control level.

$$
\begin{aligned}
u_{t_c}(\theta) = \arg\min_{u_{t_c}} \quad & \|x_{N_c}\|_P^2 + \sum_{t_c=1}^{N_c-1} \|x_{t_c}\|_Q^2 + \sum_{t_c=1}^{N_c-1} \|y_{t_c} - y_{t_c}^{SP}\|_{QR}^2 \\
& + \sum_{t_c=0}^{M_c-1} \|u_{t_c} - u_{t_c}^{SP}\|_R^2 + \sum_{t_c=0}^{M_c-1} \|\Delta u_{t_c}\|_{R1}^2 + \sum_{t=1}^{M} \|\varepsilon_{t_c}\|_{P1}^2 \\
s.t. \quad & x_{t_c+1} = Ax_{t_c} + Bu_{t_c} + C[d_{t_c}^T, Sc_{t_c}^T, des^T]^T \\
& \hat{y}_{t_c} = Dx_{t_c} + Eu_{t_c} + F[d_{t_c}^T, Sc_{t_c}^T, des^T]^T \\
& y_{t_c} = \hat{y}_{t_c} + e \\
& e = y_{t_c=0} - \hat{y}_{t_c=0} \\
& \underline{x} \le x_{t_c} \le \overline{x}, \quad \underline{y} \le y_{t_c} \le \overline{y} \\
& \underline{u} \le u_{t_c} \le \overline{u}, \quad \underline{\Delta u} \le \Delta u_{t_c} \le \overline{\Delta u} \\
& \theta = [x_{t_c=0}^T, u_{t_c=-1}^T, d_{t_c=0}^T, s_{t_c}, des^T]^T \\
& \underline{u} z_{i,t_c} \le u_{i,t_c} \le \overline{u} z_{i,t_c}, \; u_{i,t_c} = u_{t_c} \\
& \sum_{i=1}^{NP} z_{i,t_c} = 1 \\
& -y_{t_c}^* + Pur_{min} \sum_{i \in Prod} y_{i,t_c} \le -\varepsilon_{t_c} + \mathcal{M}(1-Y), \varepsilon \in [0,1] \\
& \{y_{t_c}^{SP}, u_{t_c}^{SP}, Sc_{t_c}, Y\} \subseteq s_{t_c}, \; \forall t_c \in \{0, 1, ..., N_c - 1\}
\end{aligned}
\tag{A.12}
$$

The control parameters are determined based on heuristic MPC tuning methods, and are provided in Table A.2.

Equation A.12 is an mpQP problem, which can be solved by using the POP toolbox [161].

Closed loop validation. The developed mpMPC is validated against the high fidelity model, under a range of scheduling decisions and design options. The closed loop simulations for two reactor volumes ($V_1 = 0.4m^3$, $V_2 = 1.0m^3$), are presented in Fig. 4.1.

Table A.2: Tuning parameters for the mpMPC of the CSTR for Example 1 (Reprinted with permission from [8]).

mpMPC Design Parameters	Value
N_c	6
M_c	2
QR	$\begin{bmatrix} 10^2 & 0 & 0 \\ 0 & 10 & 0 \\ 0 & 0 & 10 \end{bmatrix}$
$R1$	50
$P1$	90
Pur_{min}	0.9
$\underline{y}$	$[0, 0, 0]^T$
$\underline{u}$	$[0, 0.5, 0, 0.55]^T$
$\underline{d}$	$[0, 0.4]^T$
$\overline{y}$	$[1, 1, 1]^T$
$\overline{u}$	$[0.5, 1, 0.55, 1]^T$
$\overline{d}$	$[500, 1.0]^T$

A.2.2 Design dependent, control-aware scheduler

High fidelity model with the mpMPC embedded. The explicit control law is integrated to the original high fidelity model. The integrated model yields the closed loop dynamics of the system that is required to formulate the scheduling problem.

Model approximation. Two approximate models are derived with the discretization time steps of the scheduler (1 h) and the controller (1 min), respectively. For this particular example, the approximate model for the scheduler is derived based on a simplified first principle mole balance as presented in Eq. A.13 instead of an input-output based system identification. The mole balance expressions yield linear expressions that are directly implemented in a scheduling problem in the form of an mpMILP.

$$\frac{dW_j}{dt} = F_j - DR_j \tag{A.13}$$

where W_j is the inventory level, F_j is the product molar flow rate at the exit of the reactor, and DR_j is the demand rate of product P_j, respectively.

Three surrogate models are identified for three distinct products via the MATLAB System

Identification Toolbox. The surrogate models take the total volumetric flow rate and reactor volume as inputs and determines the product concentration set point and reference reactant composition at the feed based on the closed loop behavior. The state space matrices and the step and impulse responses of the surrogate models are presented by Eqs A.14 - A.16. Note that the discretization time of the models are identical at 15 min.

$$
\begin{aligned}
x_{t_{sm}+1} &= \begin{bmatrix} 0.004 & -0.001 & 0.002 \\ -0.031 & -0.010 & 0.045 \\ -0.118 & -0.026 & 0.118 \end{bmatrix} x_{t_{sm}} + \begin{bmatrix} -7.2 \\ -4.7 \\ -3.1 \end{bmatrix} 10^{-4} Q_{total,t_{sm}} + \begin{bmatrix} 1.7 \\ 1.4 \\ 2.1 \end{bmatrix} 10^{-3} V \\
\hat{C}_{i,t_{sm}} &= \begin{bmatrix} 0.340 & -0.037 & 0.066 \\ 0.072 & -0.040 & 0.031 \\ 0.048 & -0.042 & 0.041 \end{bmatrix} x_{t_{sm}}, \quad i \in P
\end{aligned}
\tag{A.14}
$$

$$
\begin{aligned}
x_{t_{sm}+1} &= \begin{bmatrix} 0.045 & 0.027 & -0.012 \\ 0.089 & -0.022 & -0.035 \\ 0.027 & 0.021 & -0.092 \end{bmatrix} x_{t_{sm}} + \begin{bmatrix} 2.4 \cdot 10^{-4} & 0.130 \\ 9.0 \cdot 10^{-5} & -0.749 \\ 2.9 \cdot 10^{-6} & -0.716 \end{bmatrix} \begin{bmatrix} Q_{total,t_{sm}} \\ C^{SP}_{P_2,t_{sm}} \end{bmatrix} + \begin{bmatrix} -3.8 \\ 0.2 \\ 3.5 \end{bmatrix} 10^{-5} V \\
\hat{C}_{i,t_{sm}} &= \begin{bmatrix} 0.105 & -0.038 & -0.018 \\ 0.738 & -1.005 & -0.381 \\ 0 & 0 & 0 \end{bmatrix} x_{t_{sm}}, \quad i \in P
\end{aligned}
\tag{A.15}
$$

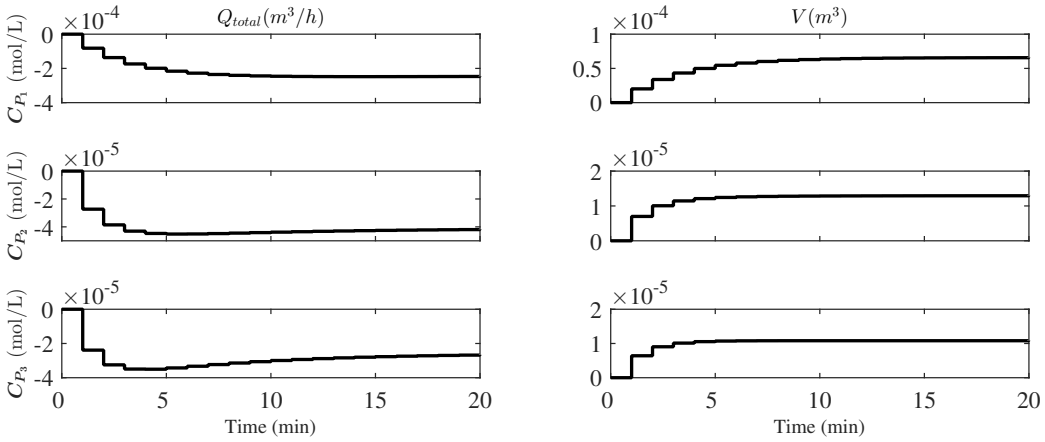


Figure A.2: Step response of Surrogate Model 1 with respect to the scheduling (Q_{total}) and design decisions (V) (Reprinted with permission from [8]).

$$
\begin{aligned}
x_{t_{sm}+1} = & \begin{bmatrix} -0.011 & -0.012 & -0.016 \\ -0.067 & 0.112 & 0.117 \\ 0.134 & -0.148 & 0.220 \end{bmatrix} x_{t_{sm}} + \begin{bmatrix} 2.5 \cdot 10^{-4} & 0.171 \\ 9.8 \cdot 10^{-5} & -0.620 \\ -5.5 \cdot 10^{-5} & 0.192 \end{bmatrix} \begin{bmatrix} Q_{total,t_{sm}} \\ C^{SP}_{P_3,t_{sm}} \end{bmatrix} \\
& + \begin{bmatrix} -2.5 \\ 0.2 \\ -1.0 \end{bmatrix} 10^{-5} V \qquad \text{(A.16)} \\
\hat{C}_{i,t_{sm}} = & \begin{bmatrix} 0.014 & -0.008 & 0.004 \\ 0 & 0 & 0 \\ 0.516 & -1.081 & 0.477 \end{bmatrix} x_{t_{sm}}, \quad i \in P
\end{aligned}
$$

where x_{t_c} is the vector of identified states. The step responses of the surrogate models are provided in Figures A.2 - A.4.

Design of the scheduler. The objective of the schedule is to minimize the inventory costs while satisfying continuous demand rate forecast within the scheduling horizon. Therefore, the objective function to be minimized is formulated as presented in Eq. A.17.

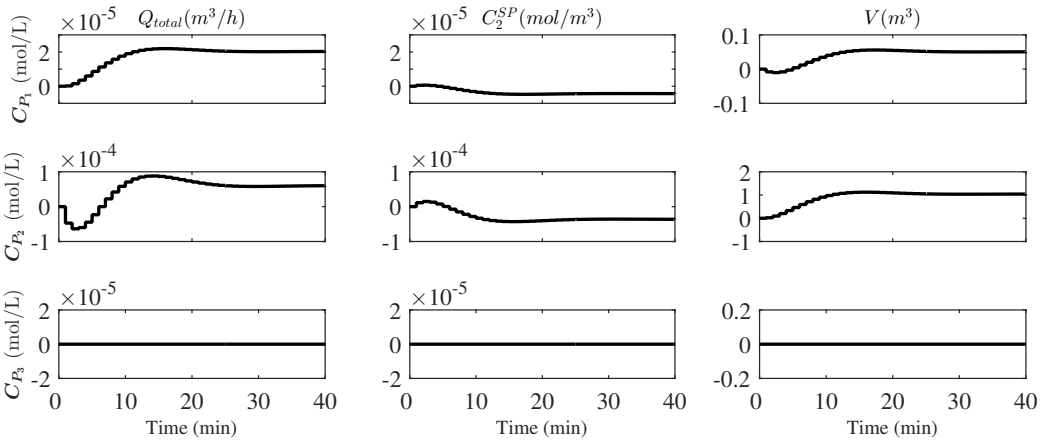


Figure A.3: Step response of Surrogate Model 2 with respect to the scheduling (C_2^{SP}, Q_{total}) and design decisions (V) (Reprinted with permission from [8]).

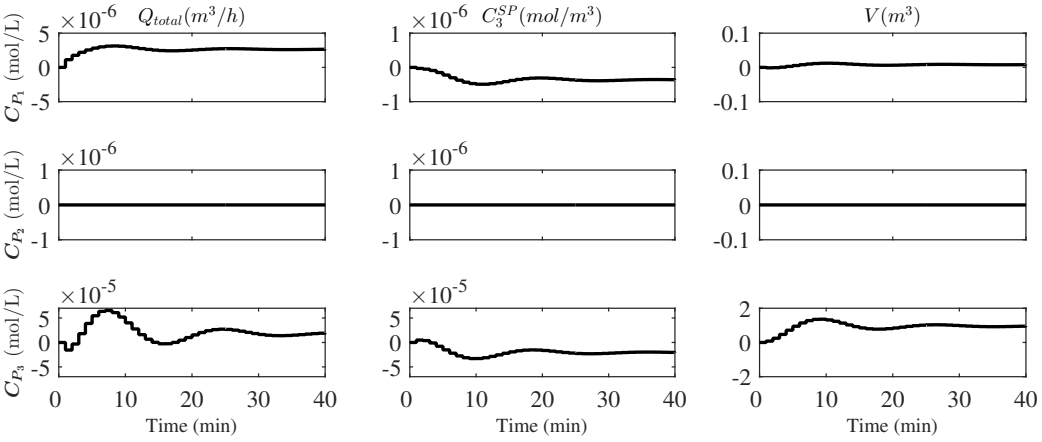


Figure A.4: Step response of Surrogate Model 3 with respect to the scheduling (C_3^{SP}, Q_{total}) and design decisions (V) (Reprinted with permission from [8]).

$$\sum_{j=1}\sum_{t=1}^{N_s}\alpha_j W_{j,t} \tag{A.17}$$

where N_s is the scheduling horizon, α_j is the storage unit cost, and $W_{j,t}$ is the inventory level of P_j at discretized time step t. This objective function is subjected to the governing dynamic approximate model given in Eq. A.13, discretized as presented in Eq. A.18.

$$W_{j,t+1} = W_{j,t} + \Delta t F_{j,t} - \Delta t DR_{j,t}, \quad \forall j, \quad \forall t \in \{1, ..., N_s - 1\} \tag{A.18}$$

The reactor is allowed to produce one product at a given time instance. Therefore, product assignment constraints are employed to enforce the system to select only one product at a time, as presented in Eq. A.19.

$$\begin{aligned} &\sum_{j=1} y_{j,t} = 1 \\ &\underline{F} y_{j,t} \leq F_{j,t} \leq \overline{F} y_{j,t} \end{aligned} \tag{A.19}$$

Capacity constraints are used to impose the physical limitations of the storage tanks, as presented in Eq. A.20.

$$\underline{W} \leq W_{j,t} \leq \overline{W} \tag{A.20}$$

The initial conditions and the demand rate forecast are defined as uncertain and bounded parameters as presented in Eq. A.21.

$$\begin{aligned} &\theta = [W_{j,t=0}, DR_{j,t}] \\ &\underline{\theta} \leq \theta \leq \overline{\theta} \end{aligned} \tag{A.21}$$

Therefore, the overall scheduling problem is constructed to minimize Eq. A.17, subjected to Eq. A.18 - A.21. The parameters of the scheduling problem are provided in Table A.3.

Table A.3: System parameters of the scheduling problem for Example 1 (Reprinted with permission from [8]).

System Parameters	Value
N_s	3
α ($/h.mol$)	$[1.0, 1.5, 1.8]^T$
$\Delta t(min)$	60
$\overline{F}$	$[50, 50, 50]^T$
$\overline{W}$	$[50, 50, 50]^T$
$\overline{D}$	$[60, 60, 60]^T$
$\underline{F}$	$[0, 0, 0]^T$
$\underline{W}$	$[0, 0, 0]^T$
$\underline{D}$	$[0, 0, 0]^T$

Design of the surrogate model. The time scale gap between the scheduler and the controller is addressed by a quadratic objective function that minimizes the L^2 norm between the volumetric flow rate determined by the schedule and the transformed decision that is passed to the controller, as presented in Eq. A.22. An additional term is included for the slack variables that take place in the purity governing soft constraints.

$$\sum_{t=0}^{M_{sm}} \|Q_{total,t} - \tilde{Q}_{total,t}\|_{R'}^2 + \sum_{t=1}^{N_{sm}} \|\varepsilon_t'\|_{P1'}^2 \tag{A.22}$$

where $\tilde{Q}_{total,t}$ is the scheduling decision, and is defined in Eq. A.23.

$$\tilde{Q}_{total,t} = \frac{\sum_j F_{j,t}}{C_{P^*,t=0}} \tag{A.23}$$

The objective function constructed in Eq. A.22 is subjected to the approximate closed loop dynamic models given by Eqs. A.14-A.16, box constraints on the inputs, outputs, and the parameters (Eq. A.24), as well as the purity soft constraints discretized in the time steps of the surrogate model).

$$\begin{aligned} \underline{u} \leq u_t := [Q_{total,t}, C_{j,t}^{SP}, \varepsilon_t'] \leq \overline{u} \\ \underline{y} \leq y_t := [C_{j,t}] \leq \overline{y} \\ \underline{d} \leq d := [\tilde{Q}_{total,t}, des] \leq \overline{d} \end{aligned} \tag{A.24}$$

Table A.4: System parameters for the surrogate model for Example 1 (Reprinted with permission from [8]).

System parameters	Model 1	Model 2	Model 3
N_{sm}	10	10	10
M_{sm}	1	1	1
$\Delta t_{sm}\ (min)$	6	6	6
R'	10^3	$\begin{bmatrix} 10^{-4} & 0 \\ 0 & 10^{-1} \end{bmatrix}$	$\begin{bmatrix} 10^{-4} & 0 \\ 0 & 10^{-1} \end{bmatrix}$
$P1'$	10^4	10^6	10^8
$\overline{u}$	$[500, 1, 1, 1, 1, 1, 1]^T$	$[500, 1, 1, 1, 1, 1, 1]^T$	$[500, 1, 1, 1, 1, 1, 1]^T$
$\overline{y}$	$[1, 1, 1]^T$	$[1, 1, 1]^T$	$[1, 1, 1]^T$
$\overline{d}$	$[500, 1.0]^T$	$[500, 1.0]^T$	$[500, 1.0]^T$
$\underline{u}$	$[0, 0, 0, 0, 0, 0, 0]^T$	$[0, 0, 0, 0, 0, 0, 0]^T$	$[0, 0, 0, 0, 0, 0, 0]^T$
$\underline{y}$	$[0, 0, 0]^T$	$[0, 0, 0]^T$	$[0, 0, 0]^T$
$\underline{d}$	$[0, 0.4]^T$	$[0, 0.4]^T$	$[0, 0.4]^T$

Three mpQP problems are constructed for three products. The surrogate model parameters are tuned to improve the closed loop performance, and are provided in Table A.4.

Closed loop validation of the integrated scheduling and control scheme. The controller, surrogate model, and the scheduler are operated simultaneously on the high fidelity model under a range of design options and product demand variations. Figure 4.2 showcases the closed loop profiles for 12 hours at the lower bound ($V_1 = 0.4m^3$) and the upper bound ($V_2 = 1.0m^3$) of the design range. After validating that the offline operational strategies on the original high fidelity model, we take the next step to formulate the MIDO problem.

Design optimization. The validated offline scheduling and control strategies are embedded in the overall MIDO problem given by Eq. 1.11 in the gPROMS environment. The capital investment-ment for the reactor is determined by Eq. A.25 [197].

$$C_e = a + bV^n \tag{A.25}$$

where C_e is the annualized reactor cost, and a, b, n are cost parameters for year 2010 given in Table A.5. The cost is escalated from 2010 to 2018 by using Eq. A.26 and the Chemical Engineering Plant Cost Index (CEPCI) [256]. The minimum total annual cost is found as $\$330k/yr$

Table A.5: CSTR cost parameters for year 2010 (Reprinted with permission from [8]).

Parameter	Value
a	61500
b	32500
n	0.6

at $V = 0.69m^3$.

$$Cost_{2018} = Cost_{2010}\frac{CEPCI_{2018}}{CEPCI_{2010}} \tag{A.26}$$

where the cost indexes $CEPCI_{2010}$ and $CEPCI_{2018}$ are 532.9 and 588.0, respectively.

A.3 Complete MIDO formulation in Chapter 5

Here, we present the complete mathematical formulation of the integrated process design, scheduling, and control problem by Eq. A.27 as an MIDO. Note that using multi-parametric programming, the original bilevel formulation presented by Eq. 5.1 is reduced to a single level dynamic optimization problem, which can be solved by existing approaches. In this study, we use ortogonal collocation on finite elements to acquire a finite dimensional MINLP problem, which is solved by a global optimization solver, BARON [221].

$$\min_{\mathbf{u}(t),\mathbf{s}(t),\mathbf{des}} \int_0^\tau C(\mathbf{x}(t),\mathbf{y}(t),\mathbf{u}(t),\mathbf{s}(t),\mathbf{des},\mathbf{d}(t))dt \tag{A.27}$$

$s.t.$ *Dynamic high fidelity model (DAE)* (A.28)

$$\begin{aligned} &\dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}(t),\mathbf{u}(t),\mathbf{s}(t),\mathbf{des},\mathbf{d}(t)), \quad \mathbf{x}(0) = \mathbf{x}_0 \\ &\underline{\mathbf{y}} \le \mathbf{y}(t) = \mathbf{g}(\mathbf{x}(t),\mathbf{u}(t),\mathbf{s}(t),\mathbf{des},\mathbf{d}(t)) \le \overline{\mathbf{y}} \end{aligned} \tag{A.29}$$

Box constraints for the variable bounds (A.30)

$$\begin{aligned} &\underline{\mathbf{u}} \le \mathbf{u}(t) \le \overline{\mathbf{u}}, \quad \underline{\mathbf{s}} \le \mathbf{s}(t) \le \overline{\mathbf{s}} \\ &\underline{\mathbf{x}} \le \mathbf{x}(t) \le \overline{\mathbf{x}}, \quad \underline{\mathbf{des}} \le \mathbf{des} \le \overline{\mathbf{des}}, \quad \underline{\mathbf{d}} \le \mathbf{d}(t) \le \overline{\mathbf{d}} \end{aligned} \tag{A.31}$$

Scheduling constraints based on SEN representation (A.32)

$$\sum_{s\in\mathcal{S}} \boldsymbol{\xi}_{j,s,t} \leq 1, \quad \forall j \in \mathcal{J}, \forall t \in \mathcal{T}$$

$$\sum_{j\in\mathcal{J}} \boldsymbol{\xi}_{j,s,t} \leq 1, \quad \forall s \in \mathcal{S}, \forall t \in \mathcal{T}$$

$$\boldsymbol{\xi}_{j,s,t+1} \leq \boldsymbol{\xi}_{j,s,t}, \quad \forall j \in \mathcal{J}, \forall s \in \mathcal{S}, \forall t \in \mathcal{T}, t \neq t_f$$

$$\mathbf{E}_{c,t} = \mathbf{E}_{c,t-1} + \sum_{j\in\mathcal{J}} \Delta\mathbf{E}_{j,c,t}, \quad \forall c \in \mathcal{C}, \forall t \in \mathcal{T}, t > 0$$

$$\sum_{s\in\mathcal{S}} \boldsymbol{\xi}_{j,s,t}\underline{V} \leq \mathbf{V}_{j,t} \leq \sum_{s\in\mathcal{S}} \boldsymbol{\xi}_{j,s,t}\overline{V}, \quad \forall s \in \mathcal{S}, \forall t \in \mathcal{T} \tag{A.33}$$

$$\mathbf{x}_s^* \leq \mathbf{x}_{s,t+1} + \mathcal{M}(\mathbf{w}_{s,t}), \quad \forall s \in \mathcal{S}^*, t = 0$$

$$\mathbf{x}_s^* \leq \mathbf{x}_{s,t+1} + \mathcal{M}\big(1 - (\mathbf{w}_{s,t} - \mathbf{w}_{s,t+1})\big), \quad \forall s \in \mathcal{S}^*, \forall t \in \mathcal{T}, 0 \leq t \leq t_f$$

$$(t+1)\mathbf{w}_{s,t} \leq \sum_{j'\in\mathcal{J}} \sum_{t'\in\mathcal{T}} \boldsymbol{\xi}_{j',s,t'}, \quad \forall s \in \mathcal{S}, \forall t \in \mathcal{T}$$

$$\boldsymbol{\xi}_{j,s^+,t} \leq \sum_{t'=0}^{t} \boldsymbol{\xi}_{j,s^-,t'}, \quad \forall j \in \mathcal{J}, \forall s^- \in \mathcal{S}^-, \forall s^+ \in \mathcal{S}^+, \forall t \in \mathcal{T}$$

$$\mathbf{x}_s^* \leq \mathbf{x}_{s,t+1} + \mathcal{M}\big(1 - \mathbf{w}_{s,t}\big), \quad \forall s \in \mathcal{S}^*, t = t_f$$

Explicit model predictive control (mpMPC) (A.34)

$$\mathbf{u}_k(\boldsymbol{\theta}) = K_{n_2}\boldsymbol{\theta} + r_{n_2}, \quad \forall \boldsymbol{\theta} \in CR_{n_2}$$

$$CR_{n_2} \coloneqq \{\boldsymbol{\theta} \in \Theta \mid A_{n_2}^{CR}\boldsymbol{\theta} \leq b_{n_2}^{CR}\}, \quad \forall n_2 \in NC_2 \tag{A.35}$$

Integrating mpMPC in the dynamic model (A.36)

Table A.6: A numerical example to encode the disjoint constraint set via base-2 numeral system.

constraint	y_1	y_2
$g_1 \leq 0$	0	0
$g_2 \leq 0$	0	1
$g_3 \leq 0$	1	0
$g_4 \leq 0$	1	1

$$
\begin{aligned}
&-\mathcal{M}\left(\sum_{i\in\{m|\beta_{n_2,m}=0\}} \bar{y}_{i,t}^{CR} + \sum_{i\in\{m|\beta_{n_2,m}=1\}} (1-\bar{y}_{i,t}^{CR})\right) \leq u_t - K_{n_2}\theta_t - r_{n_2} \\
&u_t - K_{n_2}\theta_t - r_{n_2} \leq \mathcal{M}\left(\sum_{i\in\{m|\beta_{n_2,m}=0\}} \bar{y}_{i,t}^{CR} + \sum_{i\in\{m|\beta_{n_2,m}=1\}} (1-\bar{y}_{i,t}^{CR})\right) \\
&A_{n_2}^{CR}\theta_t - b_{n_2}^{CR} \leq \mathcal{M}\left(\sum_{i\in\{m|\beta_{n_2,m}=0\}} \bar{y}_{i,t}^{CR} + \sum_{i\in\{m|\beta_{n_2,m}=1\}} (1-\bar{y}_{i,t}^{CR})\right) \\
&\sum_{i\in\{m|\beta_{n_2,m}=1\}} \bar{y}_{i,t}^{CR} - \sum_{i\in\{m|\beta_{n_2,m}=0\}} \bar{y}_{i,t}^{CR} \leq \mid m|\beta_{n_2,m}=1 \mid -1, \quad t\in\mathcal{T} \\
&\forall n_2 \in NC_2, \forall t \in \mathcal{T}
\end{aligned}
\tag{A.37}
$$

Note that the explicit form of the model predictive control (mpMPC) is derived using multi-parametric programming, which allows for reformulating the bilevel program into a single level optimization problem. Also note that the binary variables **y** from Eqs. 5.2-5.10 is substituted with $\boldsymbol{\xi}$ here to differentiate with the output **y** from the high fidelity model.

A.4 Numerical example for modeling with the base-2 numeral system

Let $g_i(x) \leq 0$ be the set of disjunctive inequalities with $i \in \{1,2,3,4\}$. Using the base-2 numeral system, we can use the combinations of two binary variables, y_1 and y_2, to uniquely activate or relax a constraint $g_i(x)$. In Table A.6, we present the binary coding that activates the corresponding constraint and relaxes the remaining constraints. Therefore, we can use the following reformulations to relax the inequalities.

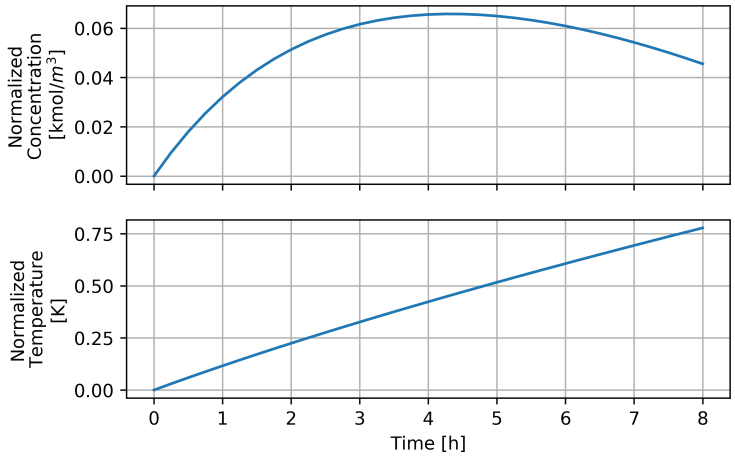


Figure A.5: Step response of the approximate model used in Example 1.

$$g_1 \leq \mathcal{M}y_1 + \mathcal{M}y_2$$

$$g_2 \leq \mathcal{M}y_1 + \mathcal{M}(1 - y_2)$$

$$g_3 \leq \mathcal{M}(1 - y_1) + \mathcal{M}y_2$$

$$g_4 \leq \mathcal{M}(1 - y_1) + \mathcal{M}(1 - y_2)$$

However, if the constraint set has three elements instead of four, i.e. $i \in \{1, 2, 3\}$, then we need to use an integer cut to exclude the combination $y_1 = 1, y_2 = 1$.

$$g_1 \leq \mathcal{M}y_1 + \mathcal{M}y_2$$

$$g_2 \leq \mathcal{M}y_1 + \mathcal{M}(1 - y_2)$$

$$g_3 \leq \mathcal{M}(1 - y_1) + \mathcal{M}y_2$$

$$y_1 + y_2 \leq 1$$

A.5 Step responses of the approximate models for the batch reactions

The step response of the approximate model used in Example 1. The same approximate model was also used in Example 2 to represent the dynamics of Reaction 1.

The approximate state space model for Reaction 2 has the following coefficient matrices, and

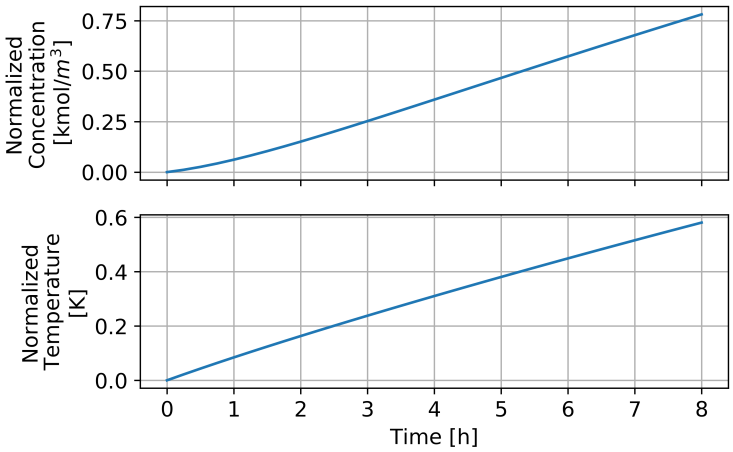


Figure A.6: Step response of the approximate model used in Example 2, Reaction 2.

the step response of the model is presented in Fig. A.6.

$$A = \begin{bmatrix} 0.9797 & -0.0301 \\ -0.0684 & 0.8518 \end{bmatrix}$$

$$B = \begin{bmatrix} -0.0056 \\ -0.0024 \end{bmatrix}$$

$$D = \begin{bmatrix} -4.2018 & 5.0875 \\ -4.0276 & 0.3175 \end{bmatrix}$$

The approximate state space model for Reaction 3 has the following coefficient matrices, and

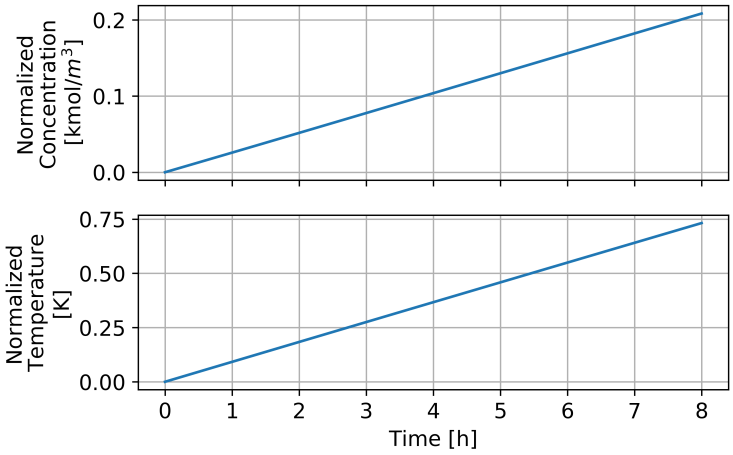


Figure A.7: Step response of the approximate model used in Example 2, Reaction 3.

the step response of the model is presented in Fig. A.7.

$$A = \begin{bmatrix} 0.9863 & 0.0418 \\ 0.0195 & 0.9385 \end{bmatrix}$$

$$B = \begin{bmatrix} 0.0082 \\ 0.0019 \end{bmatrix}$$

$$D = \begin{bmatrix} 0.6002 & 0.7780 \\ 2.4079 & 1.6798 \end{bmatrix}$$

A.6 Numerical example presented in Section 6.4

The multiparametric quadratic programming problem used as the motivating example is defined as follows.

$$\begin{aligned} z^*(\theta) = \min_x \quad & (Qx + H\theta + c)^T x \\ s.t. \quad & Ax \leq b + F\theta \\ & \underline{x} \leq x \leq \bar{x} \\ & \underline{\theta} \leq \theta \leq \bar{\theta} \end{aligned} \tag{A.38}$$

$$Q = \begin{bmatrix} 24.97 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 4 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 4.72 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1.1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.11 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1.38 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 3.24 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1.3 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 14.37 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 759.27 \end{bmatrix} \quad H = \begin{bmatrix} 0 & 0 \\ 1 & 1 \\ 0 & 1 \\ 0 & -1 \\ 0 & 0 \\ -1 & -1 \\ 0 & 0 \\ -1 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}$$

$$c = \begin{bmatrix} 5 \\ -3 \\ 1 \\ -2 \\ 2 \\ 3 \\ 4 \\ 3 \\ 5 \\ 1 \end{bmatrix}$$

$$A = \begin{bmatrix} 0.14 & 0.21 & -0.33 & 0.2 & 0 & -0.52 & -0.24 & 0 & -0.5 & 0.26 \\ -0.64 & -0.05 & -0.41 & -0.24 & 0 & -0.3 & 0.3 & 0.14 & 0 & 0.21 \\ -0.57 & -0.12 & 0 & 0 & 0 & 0 & 0.31 & 0 & 0 & -0.54 \\ 0 & 0 & 0.07 & -0.69 & 0 & 0.36 & 0.44 & 0.17 & 0 & 0.25 \\ 0.35 & 0 & -0.21 & 0.55 & 0 & -0.18 & 0 & -0.56 & 0.19 & 0.03 \\ 0 & -0.3 & -0.59 & -0.38 & 0 & 0 & 0 & -0.640 & 0 & \\ 0.08 & -0.63 & -0.45 & 0 & -0.33 & -0.25 & 0 & -0.23 & 0.15 & 0 \\ 0.24 & -0.43 & -0.17 & 0.28 & -0.51 & -0.23 & -0.36 & 0.34 & 0 & 0.28 \\ 0 & 0 & 0 & -0.17 & -0.03 & 0.44 & 0 & 0.22 & -0.72 & 0.19 \\ 0 & 0 & -0.61 & -0.28 & 0 & 0.3 & 0.41 & -0.39 & -0.37 & 0 \\ -0.14 & 0 & -0.53 & 0 & 0.14 & 0.11 & 0 & -0.76 & -0.29 & 0.026 \\ 0 & -0.44 & -0.02 & 0 & -0.39 & 0 & -0.17 & 0.4 & -0.51 & 0 \\ 0 & 0.21 & -0.11 & -0.66 & -0.14 & 0.43 & 0.15 & 0.47 & -0.05 & -0.25 \\ 0 & -0.01 & -0.37 & -0.35 & -0.29 & -0.01 & 0.36 & 0.02 & 0 & -0.17 \\ 0 & 0.32 & 0 & 0 & 0 & -0.55 & 0.34 & 0.44 & -0.24 & 0.36 \end{bmatrix}$$

$$b = \begin{bmatrix} 7.57 \\ 10.72 \\ 5.47 \\ 9.26 \\ 12.32 \\ 8.47 \\ 5.22 \\ 6.89 \\ 3.70 \\ 6.22 \\ 8.43 \\ 4.74 \\ 3.74 \\ 3.05 \\ 5.68 \end{bmatrix} \quad F = \begin{bmatrix} -0.38 & 0 \\ 0 & -0.34 \\ 0.46 & -0.25 \\ -0.33 & 0 \\ 0.38 & -0.01 \\ 0 & 0 \\ 0 & 0.39 \\ -0.11 & 0 \\ 0.41 & 0 \\ 0 & -0.04 \\ 0 & 0.09 \\ 0 & -0.45 \\ 0 & 0.06 \\ 0.59 & -0.38 \\ -0.27 & -0.15 \end{bmatrix} \quad \underline{\mathrm{x}} = \begin{bmatrix} -1E7 \\ -1E7 \\ -1E7 \\ -1E7 \\ -1E7 \\ -1E7 \\ -1E7 \\ -1E7 \\ -1E7 \\ -1E7 \end{bmatrix} \quad \bar{x} = \begin{bmatrix} 1E7 \\ 1E7 \\ 1E7 \\ 1E7 \\ 1E7 \\ 1E7 \\ 1E7 \\ 1E7 \\ 1E7 \\ 1E7 \end{bmatrix} \quad \underline{\theta} = \begin{bmatrix} -10 \\ -10 \end{bmatrix}$$

$$\bar{\theta} = \begin{bmatrix} 10 \\ 10 \end{bmatrix}$$

APPENDIX B

INTEGRATED DATA-DRIVEN PROCESS MONITORING AND EXPLICIT FAULT-TOLERANT CONTROL [1]

In this part of the Appendix, we integrate multiparametric model predictive control (mpMPC) with our previously introduced data-driven process monitoring framework [257, 258, 258] for a parametric fault-tolerant control (FTC) design framework. The developed framework can replace the conventional approach, online controller parameter re-tuning, and be used as a novel corrective maintenance strategy that significantly minimizes the process downtime spent under faulty conditions by storing pre-calculated control laws. By using multi-parametric programming [130, 259], we are able to establish the control actions for the faulty state explicitly and generate a priori, offline, maps of approximate control actions to be implemented in the online phase. This is an active fault-tolerant control strategy, specifically model-switching based active FTC, where we need to use online fault detection and identification (FDI) mechanism to monitor the process and get information on faults for further fault accommodation. Although switching based active fault-tolerant control strategies that use multiparametric programming have been introduced in the literature [260, 261, 262], the major challenge has remained to have a reliable and robust FDI system which can provide accurate fault information and minimize the number of false-alarms. Thus, we build data-driven fault detection and diagnosis models via Support Vector Machine (SVM)-based feature selection algorithm [263, 257, 264], and develop data-driven models for fault magnitude estimation via Random Forest algorithm. The developed control strategies are piecewise affine functions of the system states and the magnitude of the detected fault which are transferred to the controller via the built machine learning-based fault detection and identification (i.e. magnitude estimation) mechanism. The premise of the presented framework is to increase process resilience and

[1] Portions of this Appendix have been adapted from Onel, M., Burnak, B., Pistikopoulos, E. N., Integrated Data-Driven Process Monitoring and Explicit Fault-Tolerant Multiparametric Control, Industrial & Engineering Chemistry Research 2020, 59 (6), pp. 2291-2306. Copyright (2020) American Chemical Society.

minimize process downtime while maintaining a safe and profitable operation by enabling rapid switches between a priori mapped control action strategies. The results are presented through a semi-batch process for penicillin production.

The rest of the chapter is organized as follows: Section B.1 introduces the adopted benchmark semi-batch process. Section B.2 describes the details of the parametric fault-tolerant control design framework. Section B.3 reports the application of the framework on two distinct fault types. Finally Section B.4 provides the conclusion of the presented work.

B.1 Benchmark Semi-batch Process: Penicillin Production

We adopt fed-batch penicillin production process based on the PenSim benchmark model[265] (Figure B.1). The process operates in two modes. First, it starts in batch mode with high substrate (glucose) concentrations stimulating biomass growth. After the initial glucose level is depleted in the bioreactor, the process switches to fed-batch mode where low but non-zero glucose concentration is provided. Then, under these stressful conditions penicillin production is triggered via biomass [265, 266]. In this work, we simulate process data for fed-batch penicillin production by using the RAYMOND simulation package [267]. We produce 25 simulations for each fault magnitude and onset combinations in addition to the 200 simulations of normal operating condition (NOC) by using the RAYMOND software. Of note, fault direction is defined as $measured\ value$ - $real\ value$ within the RAYMOND simulator. In this work, a nominal feed rate of 0.06 L/h is chosen for the fed-batch phase of the simulations. A batch is terminated after a total of 30 L of substrate have been added. This corresponds to a total batch duration of approximately 549 h. The initial fermenter volume V_0, biomass concentration $C_{x,0}$, and substrate concentration $C_{s,0}$ are all independently sampled from normal distributions with mean μ and standard deviation σ. Values are limited to $\mu \pm 2.5\sigma$ in order to avoid outliers in the initial conditions. Measurements are collected from 20 process variables, where white noise is included into each of them (Table B.1). Sensors are sampled every 0.2 h which has generated an average of 2745 sample points per batch. Please note that only subset of these 20 process variables can be readily measured online in real-life and these are marked with asterisk in Table B.1. In fact, biomass, penicillin and substrate

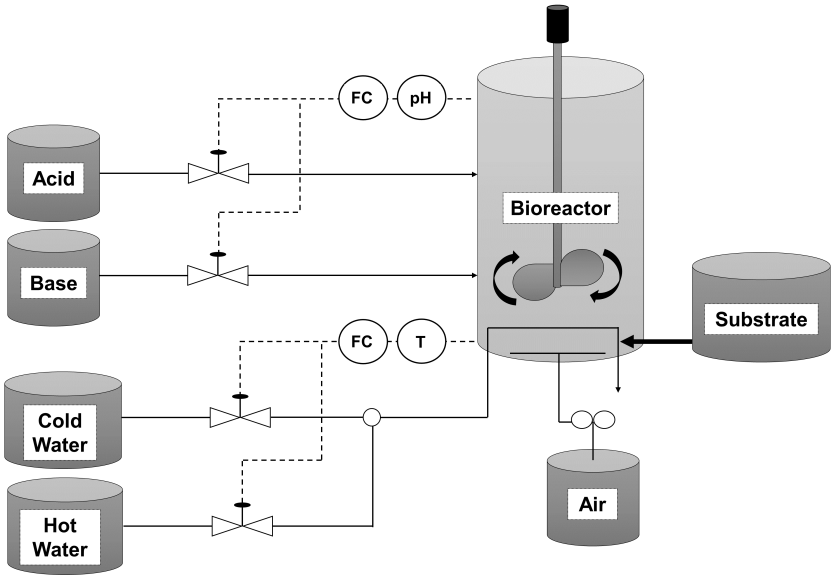


Figure B.1: Fed-batch penicillin production flow diagram (Reprinted with permission from [257]).

concentrations are indicated to be measured only offline usually every 8-10 h [266, 265]. We have utilized all 20 process variables in this work in order to demonstrate the capability of our FTC framework in handling large set of process variables.

In this work, we control the reactor temperature via fault-tolerant mpMPC by manipulating water flow rate. We have studied two distinct fault types, namely (i) sensor fault, which introduces a bias in reactor temperature measurements, and (ii) actuator fault, which creates bias in water flow rate.

B.2 Parametric Fault-Tolerant Control Framework

In order to develop a parametric fault-tolerant control system, we design a fault-tolerant mpMPC where we achieve offline optimal control strategies to be implemented for online control of the process, and build a mechanism for fault detection and magnitude estimation in the offline phase. The perspective is to obtain information on fault existence as well as magnitude of the detected fault in order to inform the mpMPC with the corrected measurements. Fault is defined as the unpermitted deviation in at least one observed variable or computed parameter of the system where controllers cannot reverse it [257]. In this work, we assume that once a fault arises in the system, it does not

Table B.1: List of process variables. Online measured variables are marked with asterisk (Reprinted with permission from [257]).

Variable Name		Initial condition	Measurement Noise (σ)	Type
1.	Substrate concentration [mg/L]	17.5 ± 1	0.01	State variable
2*.	Dissolved O_2 concentration [mg/L]	1.1601	0.004	State variable
3.	Biomass concentration [g/L]	0.1250 ± 0.030	0.5	State variable
4.	Pencillin concentration [g/L]	0	0.02	State variable
5*.	Fermentation volume [m^3]	102.5 ± 5	0.002	State variable
6*.	Dissolved CO_2 concentration [mg/L]	0.4487	0.12	State variable
7*.	pH [-]	5	0.02	State variable
8*.	Reactor temperature [K]	298	0.01	State variable (controlled)
9.	Reaction heat [cal]	0	–	State variable
10*.	Feed rate [L/h]	–	0.01	Input variable
11.	Aeration rate [L/h]	–	0.01	Input variable
12*.	Agitator power [W]	–	0.01	Input variable
13*.	Feed temperature [K]	–	0.1	Input variable
14*.	Water flow rate [L/h]	–	0.01	Input variable (manipulated)
15.	Hot/cold switch [–]	–	–	Input variable
16*.	Base flow rate [mL/h]	–	0.01	Input variable
17*.	Acid flow rate [mL/h]	–	0.01	Input variable
18.	Feed substrate concentration [mg/L]	–	–	Input variable
19.	Cooling medium temperature [K]	–	0.1	Input variable
20.	Heating medium temperature [K]	–	0.05	Input variable

fade out, therefore we need to inform the controller about the deviation in order to ensure smooth control actions.

The common first step is data acquisition. Data can be achieved via either historical operation data or process data simulations based on the dynamic model of the system, which is often readily available in industrial applications. For the offline design of the fault-tolerant mpMPC, we use normal operation data. We collect both normal and faulty operation data for building fault detection, and diagnosis and magnitude estimation models.

For fault detection, we develop two-class classification models by following simultaneous fault detection and diagnosis (s-FDD) framework [257, 264]. The major advantage of the s-FDD framework compared to the other fault detection and diagnosis (FDD) frameworks in the literature is the increased process monitoring efficiency by having one model that can detect and diagnose a fault. This creates a significant advantage during online process monitoring in terms of time efficiency where both detection and diagnosis can be achieved with a unique function evaluation. In other words, s-FDD framework yields a classification model which recognizes process abnormalities,

and marks them as process faults while it lists the major contributing process variables causing the detected fault, thus provides diagnosis. Furthermore, fault magnitude estimation models are achieved by developing regression models.

B.2.1 Offline Fault-Tolerant mpMPC Design via PAROC Framework

We build fault-tolerant mpMPC by using the PARametric Optimization and Control (PAROC) framework [130]. The steps of the framework was discussed in Chapters 2 and we follow a similar approach with the mere difference in formulating the model predictive control, which is discussed as follows.

The offline control strategy is designed to (i) track the output set points determined prior to the operation, (ii) acquire smooth control actions to maintain the longevity of the processing equipment. Therefore, the objective function of the control problem is formulated is given by the following equation.

$$\sum_{t=1}^{N} \|y_t - y_t^{sp}\|_{QR}^2 + \sum_{t=0}^{M} \|\Delta u_t - \theta^a\|_{R1}^2 \tag{B.1}$$

where N is the prediction horizon, M is the control horizon, θ^a is the magnitude of the fault acting on the corresponding actuator, $\|\cdot\|_\psi$ denotes weighted vector norm with a weight matrix ψ, QR and $R1$ are the corresponding weight matrices, and the superscript sp denotes the set point. Hence, the quadratic objective function is minimized only if the process outputs track the designated set points y^{sp}, and the consecutive control actions are smooth in the existence of faulty actuators θ^a.

The developed objective function is subjected to an approximate linear process model. However, using an approximate model to achieve closed loop control creates a mismatch between the real process outputs, y, and the predicted output values, $\hat{y}$. We address this mismatch by including Eq. B.2 in the mpMPC formulation.

$$e = y_t - \hat{y}_t, t = 0 \tag{B.2}$$

where the error term e denotes the mismatch magnitude between the real and predicted output

values at the time of measurement, $t = 0$. The error term is carried over the entire prediction horizon, as given by the equation below.

$$y_t = \hat{y}_t + e - \theta^s, \forall t \in \{1, 2, ..., N\} \tag{B.3}$$

Note that apart from the mismatch term, we also incorporate a sensor bias term θ^s to account for the sensor faults in the mpMPC. The path constraints are formulated as box constraints for the process variables to maintain certain product specifications, as presented by the equation shown below.

$$\begin{aligned} \underline{x} \leq x_t \leq \overline{x} \\ \underline{y} \leq y_t \leq \overline{y} \\ \underline{u} \leq u_t \leq \overline{u} \\ \underline{\Delta u} \leq \Delta u_t - \theta^a \leq \overline{\Delta u} \end{aligned} \tag{B.4}$$

Lastly, we define the set of parameters in the control problem as follows:

$$\theta := [x_{t=0}^T, u_{t=-1}^T, y_{t=0}^T, (y_t^{sp})^T, d_{t=0}^T, \theta^a, \theta^s]^T \tag{B.5}$$

where θ is the vector of parameters. Therefore, we postulate the explicit control strategy described by Eq. B.6.

$$\begin{aligned} u_t(\theta) = \arg\min \quad & \text{Equation B.1} \\ s.t. \quad & x^+ = Ax + Bu + Cd \\ & y = Dx + Eu + Fd \\ & B.2 - B.5 \end{aligned} \tag{B.6}$$

Note that the control strategy formulated by Eq. B.6 is a multi-parametric optimization problem with a quadratic objective function and a set of linear constraints. This class of problems can be solved exactly by using the Parametric OPtimization (POP) toolbox [161], and the solution to

these problems are expressed as a single piece-wise affine function of the parameters. Therefore, the explicit control law is derived as given by the equation below.

$$\begin{aligned} u_t(\theta) = K_i\theta + r_i, \forall t \in \{1, 2, ..., M\}, \forall \theta \in CR_i \\ CR_i := \{\theta \in \Theta \mid CR_i^A\theta \leq CR_i^b\} \end{aligned} \tag{B.7}$$

where CR denotes a polyhedral partition of the feasible parameter space, and Θ is a closed and bounded set.

Remark 7. Equation B.7 explicitly maps the exact optimal control actions for any parameter realization in Θ, if a feasible solution exists. Therefore, inclusion of the monitored faults as parameters in the explicit control law identifies the range of recovery in the existence of faulty sensors and/or actuators prior to the operation.

B.2.2 Offline Fault Detection and Reconstruction Mechanism Development

The fault detection and reconstruction mechanism is responsible from two main tasks: (i) precise and rapid fault detection and diagnosis, and (ii) accurate fault direction and magnitude estimation (a.k.a fault reconstruction). We follow the main steps of the s-FDD framework to build fault and time-specific classification models for fault detection and diagnosis. Additionally, in order to predict the magnitude of the detected fault, we develop regression models by adopting Random Forest algorithm [268]. Specifically, we regress the water flow rate measurements for the actuator fault, and reactor temperature measurements for the sensor fault. The modeling procedure for both analysis is summarized in 3 main steps. The initial step is data pre-processing which includes targeted data collection, unfolding of 3-dimensional (3D) batch process data into 2D, extracting additional process descriptors when necessary, and data scaling, respectively. This is followed by parameter tuning, and model building steps.

Data Pre-processing

Data pre-processing steps are necessary prior to model building in order to prevent bias and improve the performance of the model. Generally, data pre-processing comprises of data format-

ting, scaling and cleaning steps, where data cleaning includes both outlier removal and missing data handling. Below, we describe these three main pillars of data pre-processing in four steps. Step-1.1 to 1.3 describe data formatting, where we collect targeted process data, unfold the 3D data into 2D, and extract further features when necessary to enrich the data set. We address data scaling and cleaning steps in Step-1.4.

Targeted Data Collection: We are building (i) fault and time specific two-class C-SVM classification models for fault detection and diagnosis, and (ii) regression models for fault magnitude estimation after fault onset time. Therefore we need to gather process data around the fault onset time for both modeling. In this work, we have selected four different fault onset times, 100, 200, 300, and 400 h, where we introduce two different faults in various magnitudes. The details on the fault types and their magnitude are provided in Section B.3. In each batch, we consider the time periods that encompass the fault onset time and 10 h (50 sensor samples) afterwards, where the sensor sampling frequency is every 0.2 h.

During fault detection classifier building, we extract process data by following a sliding window approach where we receive 5 samples per hour (sensor sampling frequency of every 0.2 h). At each sensor sample, we collect historical data in 10 h blocks. For instance, to build a classifier around 100 h, we consider the time period of $100 - 110$ h of a batch. Next, starting from the fault onset time 100 h until 110^{th} h, we obtain process data in 10 h blocks: At hour 100, we collect data from $90 - 100^{th}$ h. Similarly at next sensor sample time, 100.2 h, we collect data from $90.2 - 100.2$ h. We obtain the process data iteratively until the end of the considered time period, 110 h. The schematic representation of the targeted data collection is presented in Figure B.2, wherein the gray boxes mark the fault onset time classification models are built. Blue line indicates the first, red line indicates the last 10 h data block extracted from the 90-110 h time period for 100 h fault detection classifier.. Each data collection from the selected window adds a new instance in the data set. This approach yields a 3-dimensional (3D) data with size of 2500 X 20 X 50. The first dimension of the data set is obtained with 50 sliding window iterations in 50 batches (25 faulty and 25 normal operating). Furthermore, we observe 20 process variables that includes

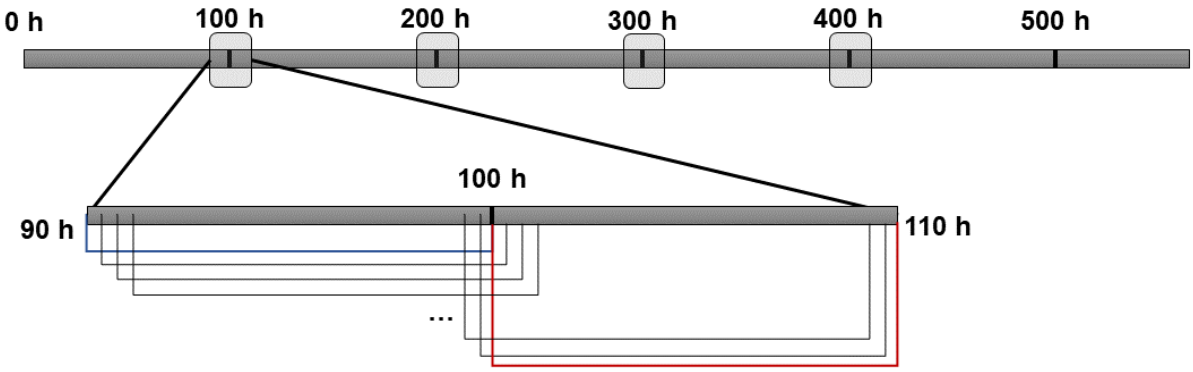


Figure B.2: Schematic representation of targeted data collection for fault detection and diagnosis classifier development (Reprinted with permission from [257]).

both state and manipulated variables B.1 in 50 sensor sample periods (i.e. $100 - 110$ h for 100 h classifier building). The data set size is consistent for each fault and time-specific fault detection model building.

On the other hand, during fault magnitude regression development, we consider solely the process variables and do not extract any further process descriptors. Here, we collect 10 h block for actuator fault, and 1 h for sensor fault magnitude estimation model development. We also combine all faulty operation data with varying fault magnitudes. Specifically, we have simulated 6 distinct fault magnitudes for sensor, and 8 for actuator faults. For each magnitude, we have simulated 25 batches. This yields 150 faulty batches with sensor fault, and 200 faulty batches with actuator fault. Next, we include equal amount of normal operating batches to our data sets. Thus, the size of the obtained data set is 300 X 20 X 5 for regression model development for sensor fault magnitude estimation. Whereas the data set size for regression model development for actuator fault magnitude estimation becomes 400 X 20 X 50. Here, the first dimension belongs to total number of batches (with equal number of faulty and normal operating batches), second dimension is the 20 process variable measurements, and the last dimension indicates the 10 h (50 sensor sample) block, and 1 h (5 sensor sample) block examined after the fault onset time of actuator, and sensor faults, respectively.

Unfolding 3D Batch Process Data into 2D: The collected 3D data needs to be unfolded into 2D prior to model building steps. The 3D data set can be unfolded in three ways by placing one out of three dimensions as rows, and grouping the other two dimensions as columns. In this work, we perform batch-wise unfolding for classification and measurement-wise unfolding for regression analysis [269]. In bath-wise unfolding, batches are the instances which are provided in the rows of the 2D data set. Whereas in measurement-wise unfolding, we keep the process variable measurements as the features and place them to the columns of the 2D data set for regression analysis. The features that constitute the columns of the 2D data sets are time-specific process variable measurements for classification, and time-specific-batch ID for regression models. After the unfolding step, the data set size becomes 2500 X 1000 for classification analysis. On the other hand, the unfolded data set size becomes 20000 X 20 for actuator fault, and 1500 X 20 for sensor fault magnitude estimation.

Extracting Additional Features: This step is optional. We apply this step only during classification analysis. The aim of this step is to enrich the data set by including additional process descriptors to capture the process nonlinearity, which can then improve classification model performances. To do this, we calculate the (i) mean, (ii) standard deviation, and (iii) slope of 20 process measurements within each sliding time window and incorporate them into the unfolded data set. This increases the classification data set sizes to 2500 X 1060.

Data Normalization & Reduction: Final data-preprocessing step is scaling of the re-configured data set and *a priori* dimensionality reduction to remove redundant features. This procedure is common to both classification and regression analysis. Each column of the 2D data set is scaled by z-score calculation, where mean of the column is extracted from each value and then divided into the standard deviation of the column. Redundant features, where the standard deviation is less than 10^{-8}, are removed in order to decrease the computational cost during offline model building phase.

Parameter Tuning

We are training (i) C-SVM (two-class) classification models by using the Gaussian radial basis function (RBF) as the nonlinear Kernel function for fault detection and diagnosis, and (ii) regression models via Random Forest algorithm for fault magnitude estimation after fault detection. Note that any regression model can be used for fault magnitude estimation, yet nonlinear regression techniques are expected to be superior than linear techniques in terms of providing more accurate fault magnitude estimations due to the nonlinear relationship between the process variables. In this work, we investigated two advanced regression techniques, namely Random Forest regression and C-parameterized Support Vector Regression (C-SVR). Specifically, we have trained C-SVR models by using the introduced feature selection algorithm in Onel et al [257] and Onel et al [264]. The results from C-SVR models are tabulated in Table S1 for actuator and Table S2 for sensor fault in the Supplementary. The results provided in Tables S1 and S2 show that dimensionality reduction does not necessarily improve the model R^2s. This is mainly due to having low ratio of the number of features to the number of instances in the process data set. Therefore, we use the entire process variables that remain after the data pre-processing step during regressor training. In this work, we prefer Random Forest algorithm over C-SVR due to the added benefit of bagging technique of Random Forest algorithm, which allows us to train more accurate regressors with the entire process variables for fault magnitude estimation. Regardless of the analysis, the initial step is parameter tuning which is required to achieve the optimal model performance.

Parameter Tuning for C-SVM Models: Here, we have two parameters to tune: (i) C (cost) parameter of C-SVM, and (ii) γ parameter of the Gaussian Radial Basis kernel function. While the first parameter acts as a regularization parameter that controls the trade-off between low training error and low test error. In other words, this parameter regulates the balance between model complexity and model generalization. Lower the training error, higher the model complexity and lower the model generalizability. On the other hand, lower the testing error, lower the model complexity and higher model generalizability but with higher training error. Finding an optimal balance is crucial to develop an accurate classifier. Furthermore, the γ parameter determines the complexity of the

Gaussian RBF kernel and affects the radius of influence of the samples selected as support vectors by the model.

In LIBSVM, the default value for RBF kernel parameter, γ, is $1/n$, where n is the number of features. Thus, we tune parameter $\hat{\gamma}$ where

$$\gamma = \frac{2^{\hat{\gamma}}}{n}. \tag{B.8}$$

Moreover, we tune parameter $\hat{C}$, where the relation between $\hat{C}$ and C is:

$$C = 2^{\hat{C}}. \tag{B.9}$$

According to the described iterative feature selection algorithm in our previous papers [257, 264], $\hat{\gamma}$ can be re-tuned after each feature elimination step with the available set of features:

$$\gamma = \frac{2^{\hat{\gamma}}}{z^T \mathbf{1}} \tag{B.10}$$

We have performed the parameter tuning via grid search and 10-fold cross-validation. In particular, we have used the values between $-1:1$ for $\hat{C}$, and $-10:10$ for $\hat{\gamma}$. We have performed the parameter tuning once in the beginning where we have the entire features in the data set. Although repeating grid search for parameters tuning after each feature elimination would be ideal, we avoid the computational cost since the attained model performance has been observed to be satisfactory. If one obtains a poor-performing model, tuning can be repeated with each available feature subsets. Finally, the parameters that produce the highest average testing accuracy are chosen for the next steps. The optimal parameters for the fault-and-time specific C-SVM models are provided in Table B.2.

Parameter Tuning for Random Forest Regression Models: In regression analysis, we have one parameter to tune, which is the number of features that can be used in training of each decision trees of the random forest model, $mtry$. This is performed via grid search among the total number of features until 1 while training Random Forest models via 10-fold cross-validation. The optimal

Table B.2: Optimal C and γ parameters of the C-SVM classifiers (Reprinted with permission from [257]).

Fault Type	Fault Onset Time	Optimal $\hat{C}$	Optimal $\hat{\gamma}$
Actuator	100 h	1	0
Actuator	200 h	1	0
Actuator	300 h	1	0
Actuator	400 h	1	0
Sensor	100 h	1	-2
Sensor	200 h	-1	0
Sensor	300 h	1	-2
Sensor	400 h	1	-2

Table B.3: Optimal $mtry$ parameters of the Random Forest regressors (Reprinted with permission from [257]).

Fault Type	Fault Onset Time	Optimal $mtry$
Actuator	100 h	12
Actuator	200 h	11
Actuator	300 h	11
Actuator	400 h	11
Sensor	100 h	16
Sensor	200 h	16
Sensor	300 h	12
Sensor	400 h	14

$mtry$ parameters are obtained by using the "trainControl" function of the "caret" package of the R statistical software. The optimal $mtry$ values for each time-specific regressors are provided in Table B.3.

Model Building

Here, we address the model building steps separately for classification and regression analysis. We follow the s-FDD framework [257, 264] to build the C-SVM classifiers for fault detection and diagnosis. The application of the framework and data-specific details are provided in Step 3.1. Furthermore, we describe the model building steps for regression analysis via Random Forest algorithm in Step 3.2.

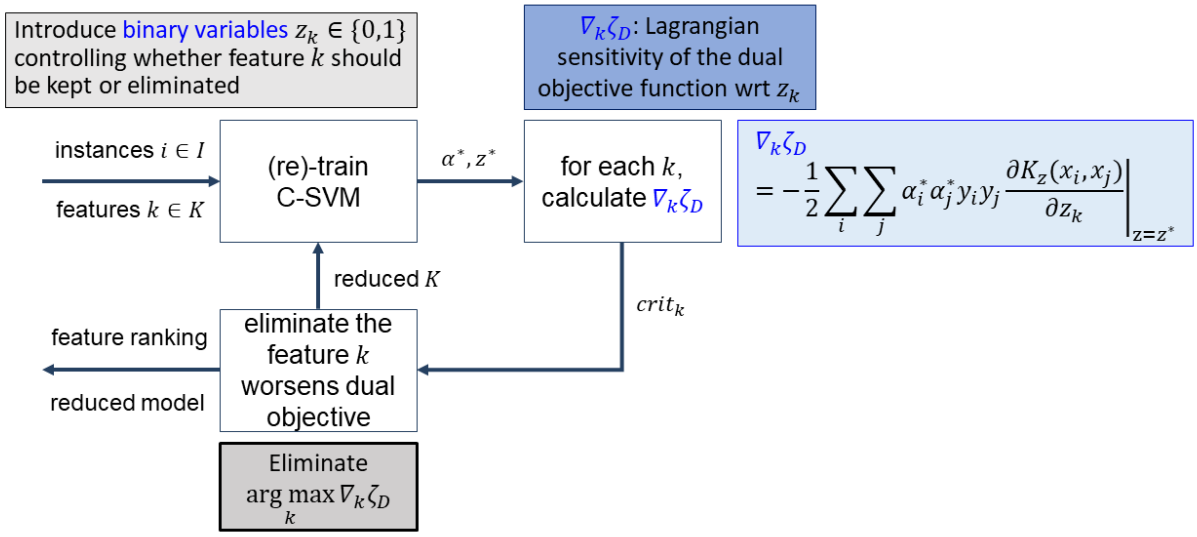


Figure B.3: Algorithmic solution procedure for simultaneous Support Vector Machine-based feature selection and modeling (Reprinted with permission from [257]).

Training C-SVM Models: The overall procedure for fault detection model development is summarized in Figure B.3.

Step-1. Feature Ranking via C-SVM Modeling: The tuned parameters are incorporated into simultaneous model-informed feature selection and classification algorithm via C-SVMs [263, 257, 264]. C-SVM binary classification models with Gaussian radial basis function (RBF) kernel are trained iteratively with each feature subset as features are eliminated one by one. Features are eliminated based on the Lagrangian sensitivity of the dual objective function of the built C-SVM model with respect to the feature subset size at each iteration. This iterative process is performed with each of the 10 train-test data set pairs which produces 10 separate feature ranking lists. Next, we create an average feature rank list based on the statistical distribution of the feature ranks among the 10 ranking lists.

Step-2. Developing C-SVM Models for each Feature Subset: Here, we re-build C-SVM models by using the optimal parameters determined in Step 2 and 10-fold cross validation, where we use the average feature rank list to guide the iterative feature elimination process. We start with the

Table B.4: C-SVM model performances. (FDR: Fault Detection Rate, FAR: False Alarm Rate) (Reprinted with permission from [257]).

Fault Type	Fault Onset Time	Accuracy (%)	AUC	FDR	FAR (%)	Optimal Feature Subset Size
Actuator	100 h	98.29	99.84	97.35	0.77	30
Actuator	200 h	98.34	99.86	97.55	0.87	35
Actuator	300 h	98.37	99.83	97.52	0.77	32
Actuator	400 h	98.77	99.05	98.29	0.75	45
Sensor	100 h	94.92	97.34	89.85	0.00	33
Sensor	200 h	98.84	99.21	97.68	0.00	59
Sensor	300 h	98.03	99.39	96.06	0.00	45
Sensor	400 h	99.00	99.38	98.00	0.00	7

whole set of features and eliminate them one by one based on this final ranking list. This process produces 10 C-SVM classifiers for each feature subset due to the 10-fold cross-validation. The performance of each model is assessed via accuracy, area under the curve (AUC), fault detection rate, and false alarm rate. We average the performance of 10 classifiers and obtain one C-SVM model performance per feature subset. At the end of this step, we tabulate the performance of C-SVM models with each feature subset. Specifically, in this work, we have generated 1060 C-SVM models.

Step-3. Choosing the C-SVM Model with Optimal Feature Subset: This step determines the final C-SVM models to be implemented in the online phase for process monitoring. Here, we select the classifier that has provided the highest model accuracy and area under the curve with minimum number of features among the 1060 C-SVM models produced in Step 4. The selected feature subset is used in analyzing the root-cause of the detected fault. Therefore, selecting minimum number of features is significant in order to facilitate the interpretation of the fault diagnosis. The performance of the selected fault-and-time specific C-SVM models are tabulated in Table B.4.

Training Random Forest Models: By using the optimal $mtry$ parameters, we train Random Forest models with 500 decision trees. Training is performed via the "randomForest" function of the "randomForest" package of R statistical software. The performance of the fault-and-time specific Random Forest models are tabulated in Table B.5.

Table B.5: Random Forest regressor performances (RMSE: Root Mean Square) (Reprinted with permission from [257]).

Fault Type	Fault Onset Time	R^2	RMSE
Actuator	100 h	0.999	0.179
Actuator	200 h	0.999	0.262
Actuator	300 h	0.999	0.248
Actuator	400 h	0.999	0.260
Sensor	100 h	0.964	0.202
Sensor	200 h	0.911	0.321
Sensor	300 h	0.985	0.129
Sensor	400 h	0.973	0.176

B.2.3 Closed-loop Validation & Online Implementation

Prior to the online implementation, we have implemented the developed fault-tolerant mpMPC, and fault detection and reconstruction mechanism to the RAYMOND simulator separately in order to validate their individual performance. The performance of fault-tolerant mpMPC is assessed by providing the fault onset time and magnitude information to the controller. We have observed that the controller adapts to the faulty condition once it is provided with accurate information on the fault type, onset time and magnitude. The accuracy of the fault detection and reconstruction mechanism is also tested and validated separately, where we have simulated a process with known fault onset and magnitude without incorporating any fault-tolerant control actions during the simulation. Finally, we implement the fault detection and reconstruction mechanism with the fault-tolerant mpMPC in the RAYMOND simulator. During the online phase, the received signals are (i) initially processed to detect the existence of any sensor or actuator faults, (ii) then reconstructed to determine the magnitude of the fault, and (iii) finally passed on to the controller for the optimal control action in the existence/absence of fault. .

B.3 Results

In this work, we control reactor temperature by manipulating water flow rate during penicillin production. We build fault-tolerant control scheme that can tolerate for both actuator and sen-

sor fault. We introduce sensor bias in water flow rate measurements for actuator fault, whereas we introduce sensor bias in reactor temperature measurements to induce sensor fault. Numerous fault magnitudes and onset time are simulated for each fault type. Particularly, we select $-2.5, -2.0, -1.5, +1.5, +2.0, +2.5$, and $-2.0, -1.5, -1.0, -0.5, +0.5, +1.0, +1.5, +2.0$ for actuator and sensor fault magnitudes during the simulations, respectively. We have developed highly accurate fault and time-specific fault detection models and regression models for fault magnitude estimation (Table B.4 and B.5) and implemented them for the fault detection and reconstruction mechanism of the established parametric fault-tolerant control system.

Figure B.4 provides a comparison of the open and closed (via fault-tolerant mpMPC) loop simulation, which signifies the importance of having an accurate control actions on the reactor temperature by manipulating the water flow rate. The mpMPC yields an offline, a priori, map of optimal control actions for the process. Figure B.5 delineates the distinct control laws for various magnitudes of sensor and actuator faults at the fixed parameters. The major advantage of the built fault-tolerant system is to gain a priori knowledge on the control actions for different fault magnitudes of actuator and sensor fault separately, as well as for different combinations of the two distinct fault types simultaneously. This map further draws the limits of the fault tolerance for each fault types. These limits indicate specific fault magnitudes for each fault type until where the designed fault-tolerant mpMPC can recover the process back to the normal condition.

From the beginning, we monitor the process with the fault tolerant mpMPC and acquire information on the existence of any fault within the system from the fault detection classifier models. In this work, the adopted alarm policy is the generation of 3 consecutive alarms. In other words, we conclude on the fault existence when we obtain 3 consecutive positive response from the fault detection classifiers. Once the fault is detected, we initiate to regress the magnitude and direction of the fault. The random forest models use the online process variable measurements to estimate the amount of deviation from the reactor temperature of the normal operating condition. Here, early detection of the faults is crucial to initiate the fault estimation process. If the fault detection latency is high, that is when fault is detected late during the operation, the controller may not be

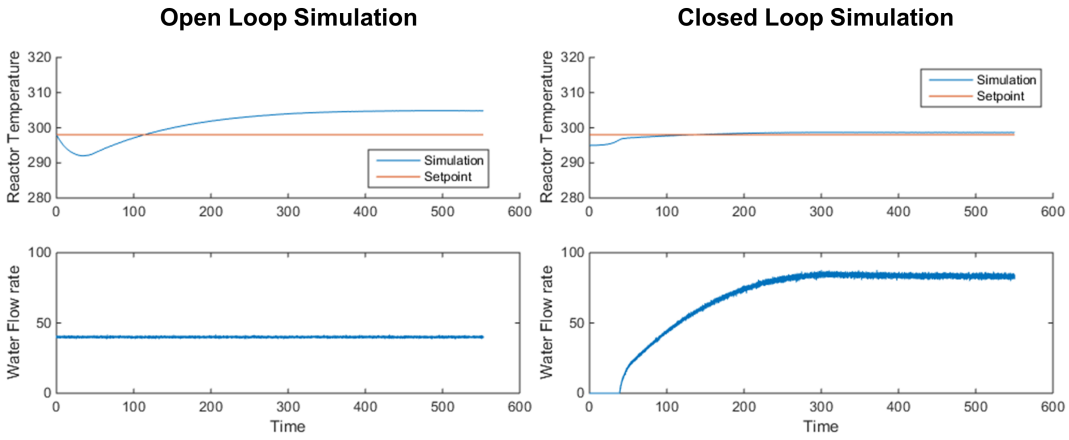


Figure B.4: Simulated reactor temperature (controlled) and water flow rate (manipulated) profiles in open and closed loop (via mpMPC) (Reprinted with permission from [257]).

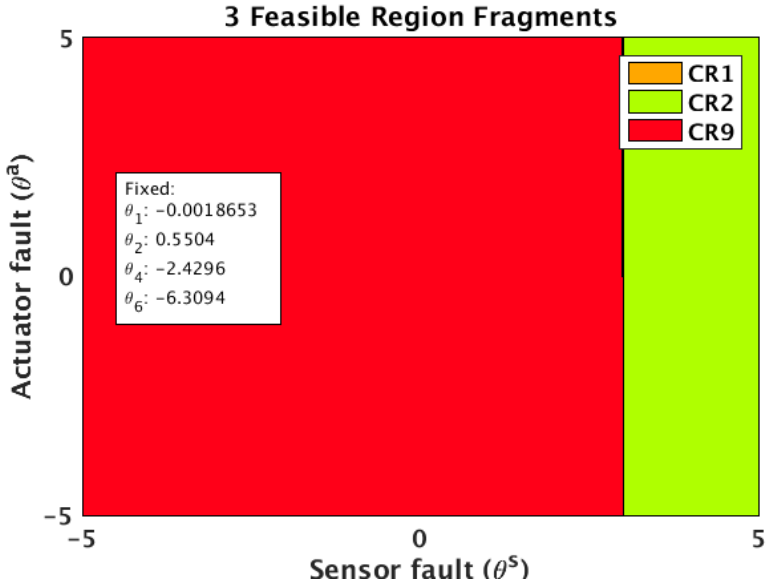


Figure B.5: A demonstration of the offline map of the fault-tolerant mpMPC strategy projected to the actuator and sensor fault magnitudes at an arbitrary time in a closed-loop simulation. Each color contains a different explicit control law as a function of the parameters. The parameters θ_1 denotes the identified state, θ_2 is the normalized process output (reacture temperature), θ_4 is the output (reacture temperature) set point, and θ_6 denotes the previous control action (Reprinted with permission from [257]).

Table B.6: Average fault detection latency of the fault&time-specific C-SVM models (Reprinted with permission from [257]).

Fault Type	Fault Onset Time	Average Latency (h)
Actuator	100 h	0.5
Actuator	200 h	1.5
Actuator	300 h	0.04
Actuator	400 h	0.16
Sensor	100 h	0.38
Sensor	200 h	1.27
Sensor	300 h	0.64
Sensor	400 h	6.17

able to return the process back to the normal condition. The reason can be two-fold: (i) the validity of the regressor may expire, thus accuracy of the fault estimation deteriorates, and (ii) there may be significant damage on the process which is irreparable. Table B.6 presents the average fault detection latency of each fault and time specific classifier among the entire simulations with varying fault magnitudes.

Achieving low latency with the fault and time-specific C-SVM models indicates early fault detection. When we compare the two different fault types, the average latency is lower for the actuator fault models. The main reason for this can be the fact that changes in water flow rate may affect the other process variables in a more definite way. This may lead to sudden changes not only in one but numerous process variables, thus facilitates the fault detection. Furthermore, the process nonlinearity affects the detection latency in distinct ways for different fault types. Specifically, we observe that we detect the actuator fault more rapidly in the later stages of the batch process, namely 300 h and 400 h models. On the other hand, the separation in the average latency is not that clear among the sensor fault detection models. Here, the high latency can be linked to the low number of process variables used in the fault magnitude estimator models, which may not be adequate to capture the process behavior in the specific process time.

We are building fault and time-specific regression models for fault magnitude estimation. Therefore, it is crucial to assess the accuracy of the fault reconstruction performance after the fault

onset time. During the online operation, we use the regressors that are trained around the simulated fault onset time. As the operation progresses after the fault onset time, where the process is kept under normal condition thanks to the fault-tolerant mpMPC, the regressor model continues to use the online process data at every new sampling point. However, as the sampling time moves away from the fault onset time, the process data characteristics can significantly change, which renders the regressor inaccurate for fault estimation. Fault estimation may not be performed as accurate as it is done near the fault onset time, which hinders the controller's learning about the process condition. This, in turn, may lead to insufficient control actions to recover the process back to the normal condition. Note that the extended validity of the regressor accuracy heavily depends on the amount of deviation of the process data characteristics. As a result, it is significant and necessary to assess the time-sensitivity of the fault estimators and identify when we need new models for accurate fault reconstruction. Furthermore, the limit of each regressor determines the targeted data collection location for the next regression model training.

Tables B.7 and B.8 tabulate the extent of the validity of the time-speicifc fault detection classifiers and magnitude estimation regressors for two set of thresholds around the reactor temperature set point being $\pm$ 0.5 and $\pm$ 0.75 K. The complete set of reactor temperature and water flow rate profiles with $\pm$ 0.5 K threshold on the set point for each time-specific models are provided in the Supplementary. In particular, we assess the extent of the validity of each time-specific model until the next time-specific model territory (i.e. 100^{th} h models are tested until 200^{th} h etc.). The results for the actuator fault case show that the models that are built at 200^{th} and 300^{th} h have been sucessfully provided necessary control actions until the target process time being 300^{th} and 400^{th} h respectively. Similarly, models built at 400^{th} h have enabled satisfactory control actions until the end of the operation. The results for the models built for 100^{th} show that the models are valid on average for the next 73.5 h and 75.4 h for $\pm$ 0.5 and 0.75 K thresholds around the reacture temperature set point, respectively. This highlights that we need to have additional models for accurate fault detection and magnitude estimation between $100^{th} - 200^{th}$ h of the batch operation.

On the other hand, for the sensor fault case, we note that the models built for 200^{th} and 400^{th}

h are not valid for an extended process time when negative fault magnitudes are observed. On average, the models are valid for another 1.3 and 1.5 h after fault is introduced in 200 and 400 h, respectively when the reactor temperature deviation threshold is set to 0.5 K. When we increase the threshold to 0.75 K around the set point, we observe that the models built at 200^{th} can maintain a smooth operation for its entire targeted operation range, which is the next 100 h, because the latency in fault detection has caused a deviation that is higher than 0.5 but lower than 0.75 K. However this does not apply to the models for 400^{th} h. The threshold increase does not extend the validity of 400^{th} h models since the maximum deviation observed is as high as 2.1 K (see the Supplementary Information by Onel et al. (2020) [229]). The reason behind the limited model validity for the two time-specific models at 200 and 400 h is due to the high fault detection latency. In other words, by the time we detect the fault occurring at 200^{th} and 400^{th} h, the deviation from the reactor temperature set point already exceeds the predetermined thresholds see the Supplementary Information by Onel et al. (2020) [229]). Therefore, required control actions are not provided by the controller as it has not been notified with the existence and magnitude of the fault. In order to overcome this problem, fault detection latency is required to be improved. This can be achieved by increasing the frequency of the fault detection classifiers between 200 and 400 h of the batch operation.

In order to provide a comparison between the two fault types, we provide the reactor temperature and water flow rate profiles for 100 h models. Particularly, we display the profiles of the simulations where we introduce actuator fault with -2.5 and $+2.5$ fault magnitude in Figures B.6 and B.7, respectively. Additionally, Figures B.8 and B.9 demonstrate the profiles of the simulations where we introduce sensor fault with -2 and $+2$ fault magnitude. The profiles with actuator fault show that once the regressor model validity expires with the altering dynamics of the batch process, the correction in the faulty water flow rate disrupts and deteriorates. This leads to a significant increase in the reactor temperature that leads to a possible system failure. On the other hand, the early capture of the sensor fault leads to rapid and necessary changes in the water flow rate which enables fast process recovery back to the normal condition. Of note, in order to ensure

Table B.7: Extent of time-specific fault detection and magnitude estimation model validity for actuator fault (Reprinted with permission from [257]).

Fault Onset Time (h)	Fault Magnitude	Validity Limit for 0.5 K threshold (h)	Validity Limit for 0.75 K threshold (h)
100	-2.5	143.2	146.6
100	-2	147.3	150.7
100	-1.5	151.2	155.0
100	1.5	199.4	200.0
100	2	200.0	200.0
100	2.5	200.0	200.0
200	-2.5	200.0	200.0
200	-2	300.0	300.0
200	-1.5	300.0	300.0
200	1.5	300.0	300.0
200	2	300.0	300.0
200	2.5	300.0	300.0
300	-2.5	400.0	400.0
300	-2	400.0	400.0
300	-1.5	400.0	400.0
300	1.5	400.0	400.0
300	2	400.0	400.0
300	2.5	400.0	400.0
400	-2.5	Through the end	Through the end
400	-2	Through the end	Through the end
400	-1.5	Through the end	Through the end
400	1.5	Through the end	Through the end
400	2	Through the end	Through the end
400	2.5	Through the end	Through the end

Table B.8: Extent of time-specific fault detection and magnitude estimation model validity for sensor fault (Reprinted with permission from [257]).

Fault Onset Time (h)	Fault Magnitude	Validity Limit for 0.5 K threshold (h)	Validity Limit for 0.75 K threshold (h)
100	-2.0	187.9	200.0
100	-1.5	187.4	200.0
100	-1.0	187.2	200.0
100	-0.5	187.2	200.0
100	0.5	187.6	200.0
100	1.0	187.6	200.0
100	1.5	187.6	200.0
100	2.0	187.4	200.0
200	-2.0	201.3	300.0
200	-1.5	201.3	300.0
200	-1.0	201.3	300.0
200	-0.5	201.3	300.0
200	0.5	300.0	300.0
200	1.0	300.0	300.0
200	1.5	300.0	300.0
200	2.0	300.0	300.0
300	-2.0	358.1	358.1
300	-1.5	356.4	356.4
300	-1.0	359.9	359.9
300	-0.5	358.0	358.0
300	0.5	369.9	369.9
300	1.0	360.5	360.5
300	1.5	360.4	360.4
300	2.0	363.1	363.1
400	-2.0	401.4	401.4
400	-1.5	401.4	401.4
400	-1.0	401.4	401.4
400	-0.5	401.5	401.5
400	0.5	Through the end	Through the end
400	1.0	Through the end	Through the end
400	1.5	Through the end	Through the end
400	2.0	Through the end	Through the end

smooth control actions, one needs to switch to the next valid model once the validity of the previous model expires. This is necessary in order to capture dynamic process characteristics and detect any possible faults. The presented simulation profiles with actuator and sensor faults prove that the designed fault-tolerant mpMPC provides smooth control actions successfully.

Finally, we perform a sensitivity analysis with the time-specific fault detection and magnitude estimation models built at 100^{th} and 200^{th} h in order to determine the perimeter of the model effectiveness. To this end, we use the time-specific models for $\pm$ 30 h perimeter of their corresponding process time. Particularly, the C-SVM model for fault detection and random forest model for the fault magnitude estimation are utilized for every 5 h fault onsets between 70^{th} and 130^{th} h with the models built at 100^{th} h and between 170^{th} and 230^{th} h with the models built at 200^{th} h (Figure B.10). We adopt the $\pm$ 0.5 K threshold around the reactor temperature set point and only utilize the extreme negative and positive fault magnitudes simulated in this work (-2.5 and +2.5 for actuator and -2 and +2 for sensor fault) for the analysis. The results reveal that actuator fault models have more limited range compared to sensor fault models. In particular, the models built at 100^{th} have successfully been used between $90 - 100^{th}$ h of the batch operation. The validity range for the models built at 200^{th} reaches to 15 and 20 h for negative and positive fault magnitudes, respectively. On the other hand, the analysis yield that the models built at 100^{th} and 200^{th} h for sensor fault have been able to perform required control actions for the analyzed 30 h perimeter except the analysis performed with negative fault magnitude with models built at 200^{th} h. This is again due to the fact that by the time fault is detected the raise in the reactor temperature exceeds the allowed region (see the Supplementary Information by Onel et al. (2020) [229]). When the deviation threshold is raised to $\pm$ 0.75 K, the time-specific models are shown to be valid for the entire analyzed 30 h perimeter (Figure B.11). This analysis is significant to elucidate the effectiveness limit of the time-specific models which is required to determine the model switching frequency during online monitoring. Overall, the results demonstrate the need for additional models during 100-200 h of the operation if a strict deviation threshold (i.e. 0.5 K) is preferred during the operation. Yet for 0.75 K deviation threshold, the presented time-specific models have successfully

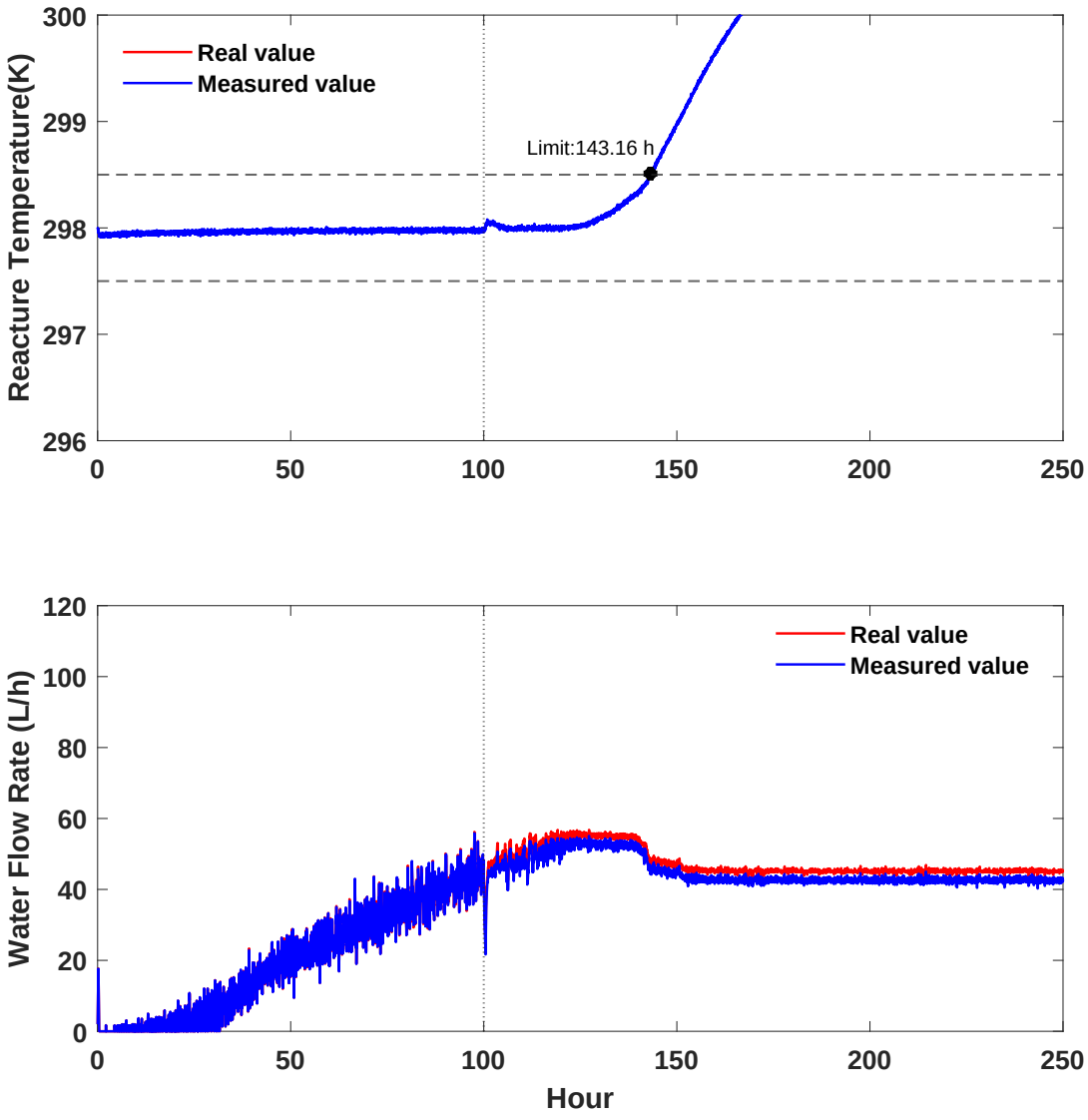


Figure B.6: Reactor temperature and water flow rate profiles for process with actuator fault. Fault Onset: 100 h, Fault Magnitude: -2.5 (Reprinted with permission from [257]).

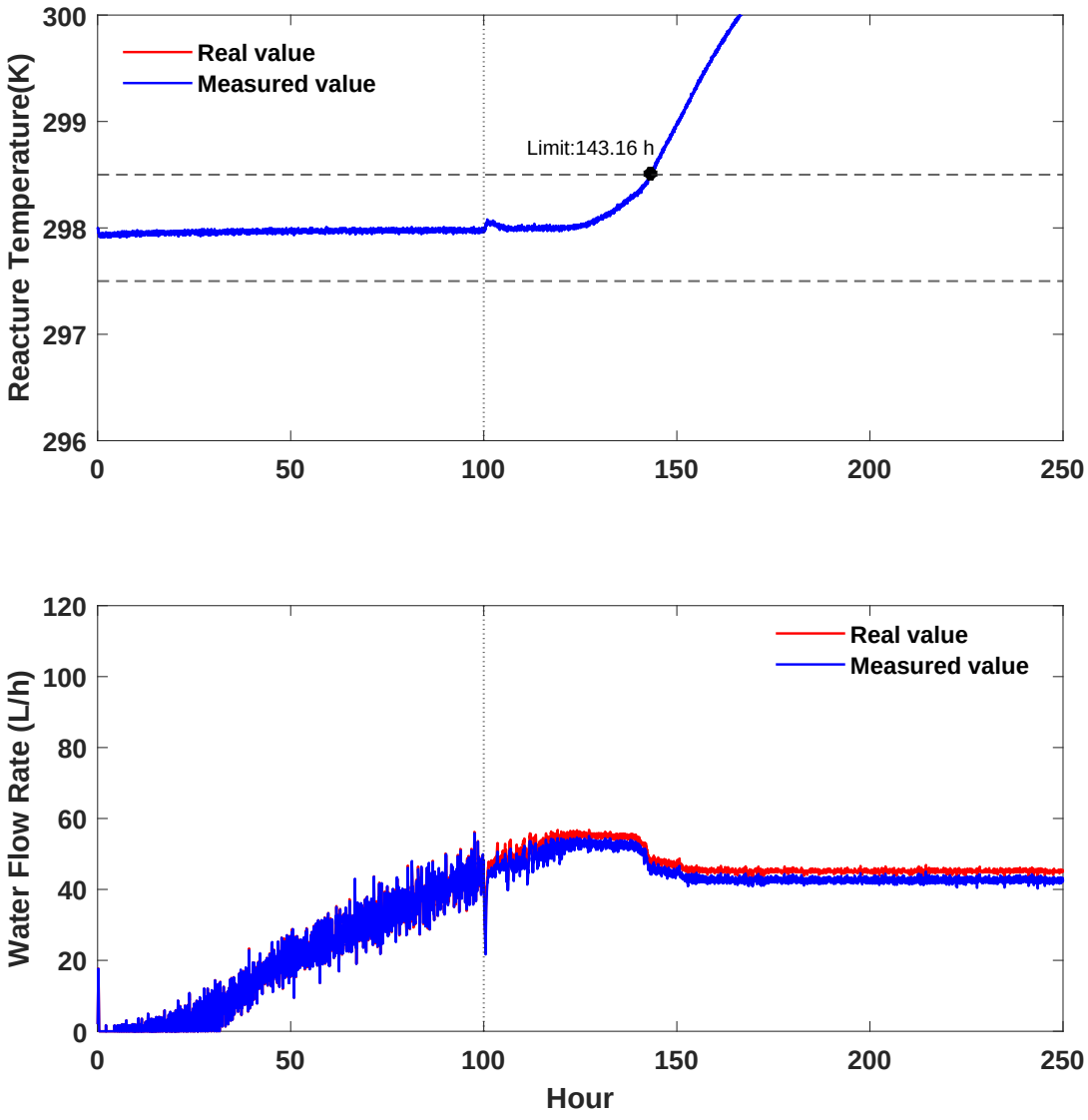


Figure B.7: Reactor temperature and water flow rate profiles for process with actuator fault. Fault Onset: 100 h, Fault Magnitude: +2.5 (Reprinted with permission from [257]).

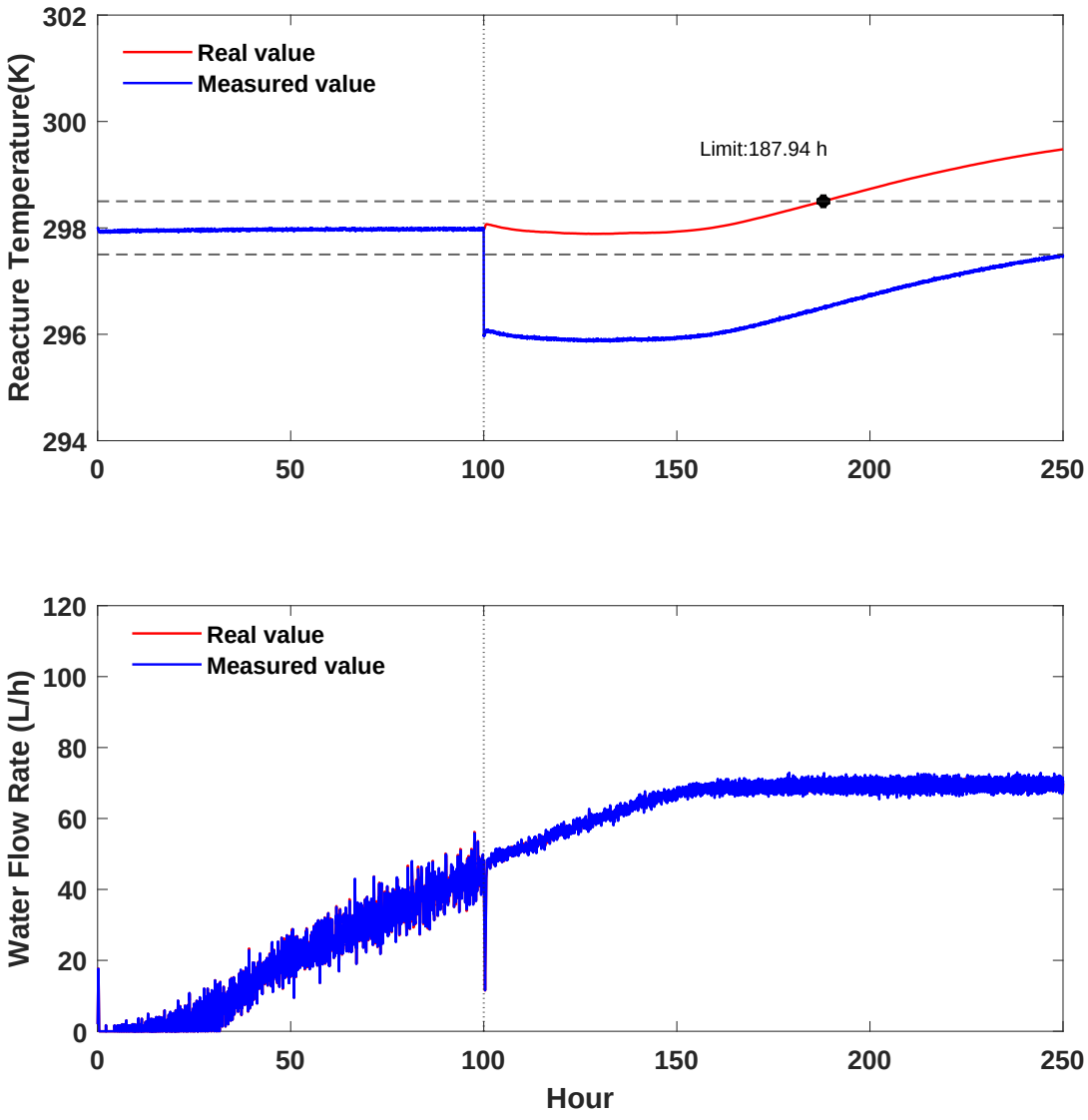


Figure B.8: Reactor temperature and water flow rate profiles for process with sensor fault. Fault Onset: 100 h, Fault Magnitude: -2.0 (Reprinted with permission from [257]).

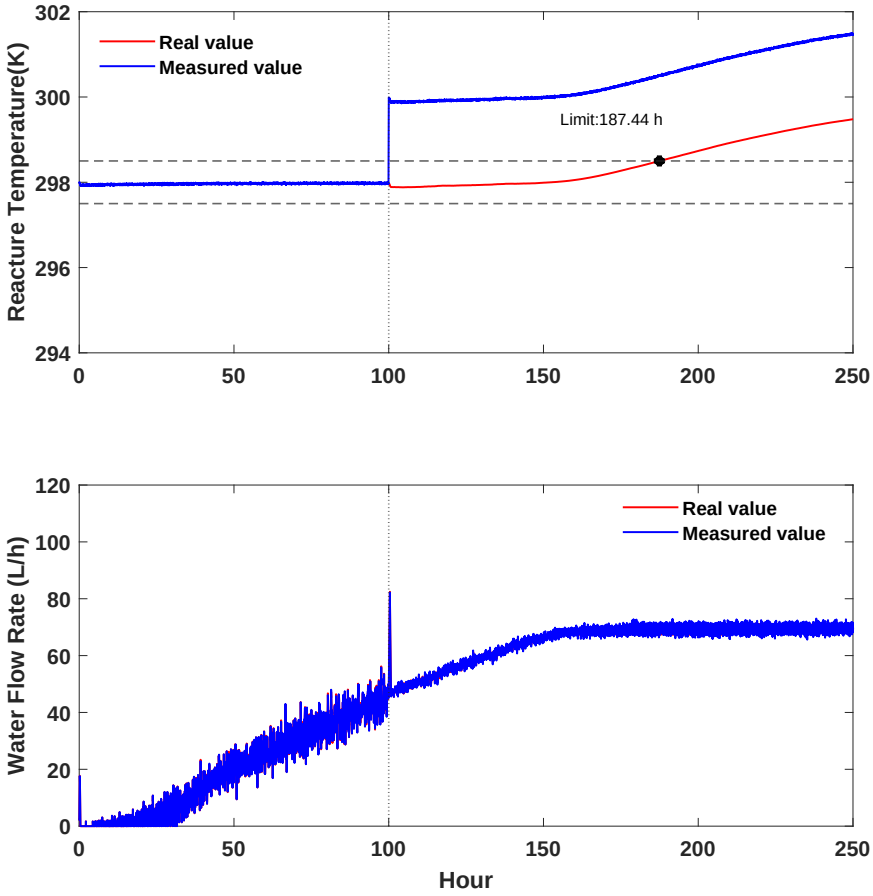


Figure B.9: Reactor temperature and water flow rate profiles for process with sensor fault. Fault Onset: 100 h, Fault Magnitude: +2.0 (Reprinted with permission from [257]).

provided satisfactory control actions under faulty condition.

B.4 Conclusions

As the effect of smart manufacturing revolution propagates and influences the vision of numerous industrial operations, development of fault-tolerant control system becomes one of the major factors in achieving high process resilience. Traditional corrective maintenance strategies include controller re-tuning which leads to longer process downtime that may adverse the end-product quality and cause higher operation cost. This work proposes a novel parametric fault-tolerant control framework that enables rapid and accurate switches within the offline map of control actions to eliminate process downtime, and maximize process reliability. This further enables attaining higher product quality which leads to higher profit from the operation.

In this work, we present a novel active fault-tolerant strategy and corrective maintenance strategy which benefits from multiparametric programming and machine learning-based process monitoring. Particularly, we have designed multiparametric model predictive controller by following the PAROC framework [130] and the s-FDD framework. The s-FDD framework is used to formulate the fault detection and reconstruction mechanism of the fault-tolerant system, where the built classifiers provide the information on fault existence and regressors yield the fault magnitude and direction estimation. The trained C-SVM models with the optimal feature subset further enables the rapid diagnosis of the detected of fault. The average accuracy of the classifiers is %98.44, and %97.70 for the actuator and sensor faults, respectively. Moreover, the average R^2 of the trained regressors is 0.999 and 0.958 for the actuator and sensor faults, respectively. The presented approach formulates as a novel active fault-tolerant strategy in which an accurate and robust fault detection and reconstruction mechanism is ensured via the s-FDD framework and multiparametric MPC enables rapid switches between fault-tolerant control actions. Please note that the presented fault-tolerant strategy is agnostic to any fault types, thus it can be extended to process faults as well. Finally, we note that the design of the fault-tolerant mpMPC can further enable the handling of simultaneous faults as it includes the deviation in both process variables (i.e. reactor temperature and water flow rate) as additional parameters.

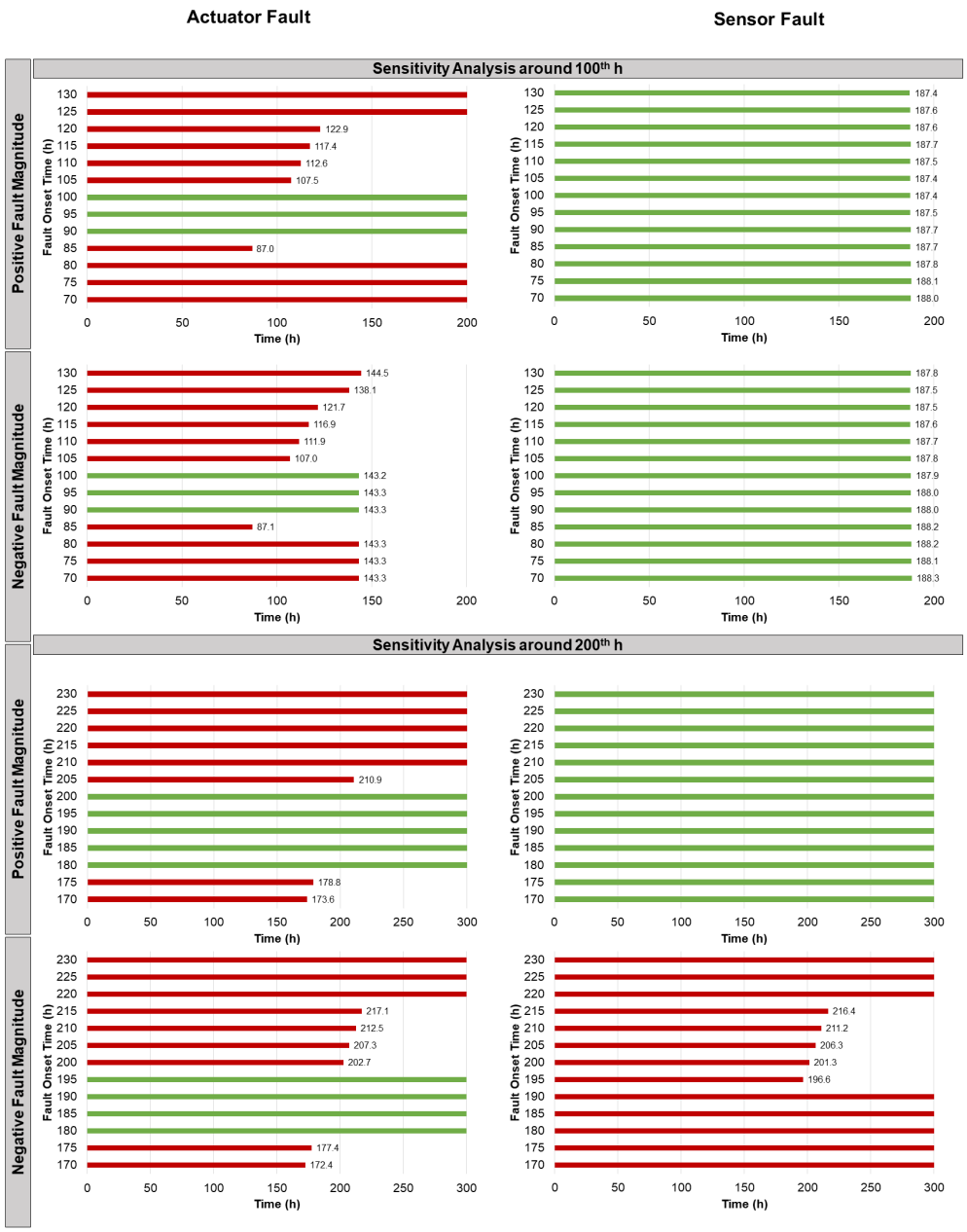


Figure B.10: Sensitivity analysis of time-specific models built at 100^{th} and 200^{th} h for actuator and sensor faults. The set point deviation threshold is $\pm$ 0.5 K. The green bars highlight that the model is satisfactorily valid. The red bars belong to limited time model validity cases. Note that once a red bar is assigned, the further hours are automatically assigned with red (Reprinted with permission from [257]).

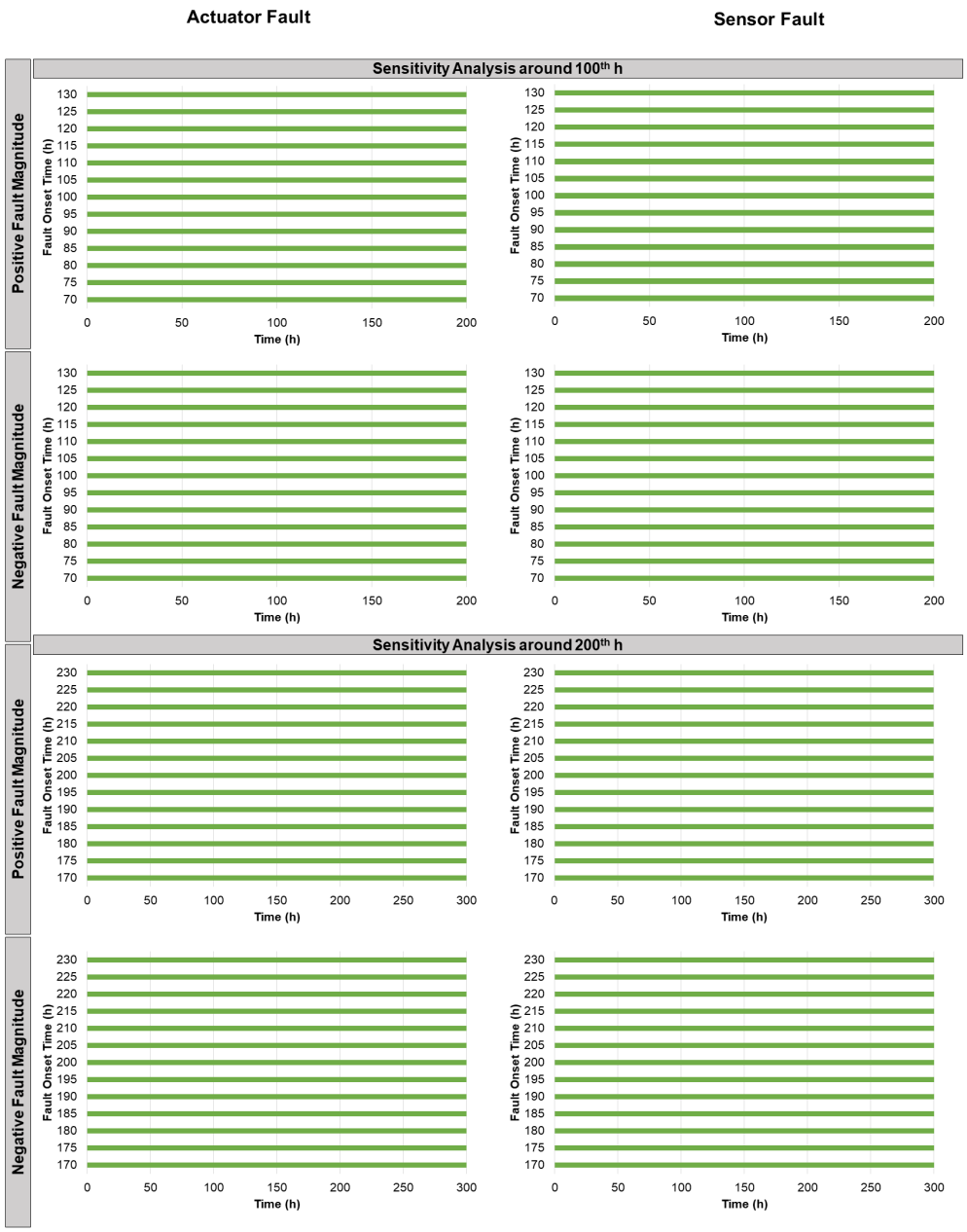


Figure B.11: Sensitivity analysis of time-specific models built at 100^{th} and 200^{th} h for actuator and sensor faults. The set point deviation threshold is $\pm$ 0.75 K. The green bars highlight that the model is satisfactorily valid. The red bars belong to limited time model validity cases. Note that once a red bar is assigned, the further hours are automatically assigned with red (Reprinted with permission from [257]).

APPENDIX C

LIST OF PUBLICATIONS

C.1 Monograph

- **Burnak, B.**, Pistikopoulos, E. N., Integrated process design and operational optimization via multi-parametric programming, Morgan & Claypool Publishers, to appear in 2020.

C.2 Journal Articles

- **Burnak, B.**, Pistikopoulos, E. N., Integrated process design, scheduling, and model predictive control of batch processes with closed-loop implementation, *in review*.

- **Burnak, B.**, Katz, J., Pistikopoulos, E.N., A novel geometrical algorithm for multiparametric programming, *in review*.

- Onel, M., **Burnak, B.**, Pistikopoulos, E. N., Integrated Data-Driven Process Monitoring and Explicit Fault-Tolerant Multiparametric Control. Industrial and Engineering Chemistry Research 2020, 59, 2291-2306.

- Tian, Y., Pappas, I., **Burnak, B.**, Katz, J., Pistikopoulos, E. N., A Systematic Framework for the Synthesis of Operable Process Intensification Systems - Reactive Separation Systems. Computers & Chemical Engineering 2020, 134, 106675.

- Tian, Y., Pappas, I., **Burnak, B.**, Katz, J., Pistikopoulos, E. N., Simultaneous Design & Control of a Reactive Distillation System - A Parametric Optimization & Control Approach, *in review*.

- **Burnak, B.**, Diangelakis, N. A., Pistikopoulos, E. N., Towards the grand unification of process design, control, and scheduling - Utopia or reality?. Processes 2019, 7 (7), 461.

- **Burnak, B.**, Diangelakis, N.A., Katz, J., Pistikopoulos, E.N., Integrated process design, scheduling, and control using multiparametric programming, Computers & Chemical Engineering 2019, 125, 164-184, *Special Issue*.

- Papathanasiou, M. M., **Burnak, B.**, Katz, J., Shah, N., Pistikopoulos, E.N., Assisting continuous biomanufacturing through advanced control in downstream purification, Computers & Chemical Engineering 2019, 125, 232-248, *Special Issue*.

- **Burnak, B.**, Katz, J., Diangelakis, N.A., Pistikopoulos, E.N., Simultaneous Process Scheduling and Control: A Multiparametric Programming-Based Approach, Industrial & Engineering Chemistry Research 2018, 57, 3963-3976.

- Katz, J., **Burnak, B.**, Pistikopoulos, E.N., The impact of model approximation in multiparametric model predictive control, Chemical Engineering Research and Design 2018, 139, 211-223.

- Diangelakis, N.A., **Burnak, B.**, Katz, J., Pistikopoulos, E.N., Process Design and Control optimization: A simultaneous approach by multi-parametric programming, AIChE Journal 2017, 63 (11), 4827-4846.

C.3 Conference Proceedings

- Tian, Y., Pappas, I.S., Katz, J., **Burnak, B.**, Avraamidou, S., Diangelakis, N.A., Pistikopoulos, E.N., Towards a systematic framework for the synthesis of operational process intensification systems - Application to reactive distillation systems., 29th European Symposium on Computer-Aided Process Engineering (ESCAPE-29) 2019, 73-78.

- Papathanasiou, M. M., **Burnak, B.**, Katz, J., Pistikopoulos, E.N., Control of a dual mode separation process via multi-parametric Model Predictive Control., 12th IFAC Syposium on Dynamics and Control of Process Systems (DYCOPS 2019), 988-993.

- Papathanasiou, M. M., **Burnak, B.**, Katz, J., Pistikopoulos, E. N., Control of Small-Scale Chromatographic Systems under Disturbances. Foundations of Computer-Aided Process Design (FOCAPD 2019), IFAC-PapersOnLine 47, 269-274.

- **Burnak, B.**, Katz, J., Diangelakis, N.A., Pistikopoulos, E.N., Integration of Design, Scheduling, and Control of Combined Heat and Power Systems: A Multiparametric Programming Based Approach, Computer Aided Chemical Engineering 2018, 44, 2203-2208.

- Diangelakis, N.A., **Burnak, B.**, Pistikopoulos, E.N., A multi-parametric programming approach for the simultaneous process scheduling and control - Application to a domestic cogeneration unit, Foundations of Computer Aided Process Operations/Chemical Process Control (FOCAPO/CPC) 2017, 8-12.

- Diangelakis, N.A., **Burnak, B.**, Pistikopoulos, E.N., Design, scheduling and control: A simultaneous approach by multi-parametric programming, Computing and Systems Technology Division 2016 - Core Programming Area at the 2016 AIChE Annual Meeting, 145-147.