Risvan Dirza

Coordinated Feedback-optimizing Control for Large Scale Processes

with applications of field-wide oil & gas production system

Thesis for the Degree of Philosophiae Doctor

Trondheim, June 2024

Norwegian University of Science and Technology
Faculty of Natural Sciences
Department of Chemical Engineering

NTNU

Norwegian University of
Science and Technology

NTNU

Norwegian University of Science and Technology

Thesis for the Degree of Philosophiae Doctor

Faculty of Natural Sciences
Department of Chemical Engineering

ISBN (printed version)
ISBN (electronic version)
ISSN 1503-8181

Doctoral theses at NTNU,

Printed by NTNU-trykk

Summary

The thesis is dedicated to propose optimization strategies tailored for complex and/or large-scale process systems, such as those found in oil and gas production operations. These systems consist of various sub-processes characterized by varying timescales and numerous constraints. Among the strategies explored, real-time optimization (RTO) stands out prominently, wherein optimization results are translated into setpoints for the control layer. Despite its potential benefits, numerical-based RTO sees limited practical application, mainly due to the cost of developing and updating process models, as well as partly too infrequent updates. Moreover, the maintenance of advanced tools poses challenges, often requiring specialized skills that are not widely available. Consequently, the industry has yet to fully exploit the potential of RTO. This thesis advocates for a pragmatic approach, leveraging common, well-known and established tools such as PID controllers and selectors, supplemented by small-scale numerical solvers for particular cases. This strategy aims to enhance the feasibility of implementation in practical settings. Experimental validation is also undertaken to substantiate the efficacy of the proposed strategies.

In the first part of this thesis, the focus is on active constraint switching within small-scale systems, which is addressed comprehensively in Chapter 2. A proposed solution lies in the development of a primal-dual framework to handle automatic active constraint switching with fixed control structure. However, challenges emerge when dealing with tight constraints and the dependency on accurate estimation of gradients, both pivotal for ensuring safety and environmental compliance. Chapter 3 suggests strategies to overcome these challenges, advocating for the integration of override constraint controllers and auxiliary constraints within the primal-dual framework.

In the second part, Chapter 4 presents a dual-based distributed feedback-optimizing system, augmented with and without solution predictors, designed to decompose large-scale (interactive) systems into more manageable and smaller subsystems. This approach not only facilitates the decomposition process but also enables the achievement of system-wide optimal performance through the introduction of a coordinator for these subsystems. The efficacy of this approach is demonstrated in Chapter 5, where experimental results from a lab-scale gas-lift rig validate its effectiveness. Similar to primal-dual, dual-based distributed feedback optimization relies on accurate real-time gradient estimations, yet distinguishes itself by offering enhanced insights into the effect of non-performing local gradient estimations. Consequently, Chapter 6 proposes a systematic pairing procedure for an override controller, aimed at minimizing economic losses. Moreover, to address constraints such as limited input rates and the need for joint control over a critical constraint,

Chapter 7 outlines several systematic procedures for multi input override control. Transitioning to primal-based distributed feedback-optimizing systems in the latter part of this part, Chapter 8 introduces an alternative approach adept at handling a total input constraint, and also theoretically possible for an output constraint. Finally, the experimental validation of these approaches, including primal-based and dual-based distributed feedback-optimizing systems with and without overrides, culminates in the discussion presented in Chapter 9.

The last part of the thesis addresses practical issues. For instance, while both primal- and dual-based distributed feedback-optimizing systems typically necessitate a central coordinator, implementing such frameworks in a marginal offshore field poses significant practical challenges. To tackle this issue, Chapter 10 proposes a decentralized alternative, introducing a graph-based primal-based distributed feedback-optimizing system that offers increased adaptability and efficiency. Furthermore, Chapter 11 explores extensions to gas-lift systems aimed at enhancing realism. Due to the lack of sensor reliability, the utilization of self-optimizing control is suggested.

The thesis notably highlights the imperative for implementing customized strategies to tackle the various challenges within oil and gas production systems. It emphasizes the importance of tailored approaches aimed at resolving distinct subproblems inherent to these complex systems. This various approaches underscores the necessity for targeted solutions designed to address various issues encountered within large-scale production operations.

Acknowledgements

In the name of Allah, the Most Gracious, the Most Merciful. Praise be to Allah, the Lord of all worlds, and peace and blessings be upon the Prophet Muhammad, his family, and his companions. I begin by expressing my deepest gratitude and appreciation to Allah Almighty, the source of all knowledge and wisdom, for granting me the strength, guidance, and perseverance throughout this academic journey.

This thesis would not have been completed without the help of many people. First of all, I would like to express my deepest gratitude to Prof. Sigurd Skogestad, my esteemed advisor, for his unwavering support, guidance, and intellectual stimulation throughout the entire journey of my PhD research. Prof. Skogestad's profound knowledge, insightful feedback, and commitment to excellence have been instrumental in shaping the direction of my academic journey. I am truly fortunate to have had the opportunity to work under Prof. Skogestad's mentorship. His passion for research and dedication to pushing the boundaries of knowledge have been a constant source of inspiration. His mentorship has not only enriched my academic experience but has also played a pivotal role in my personal and professional development.

I would like to thank Assistant Professor Dinesh Krishnamoorthy from Eindhoven University of Technology (TU/e) for his unwavering support, and guidance throughout my doctoral journey. His expertise, dedication, and insightful feedback have been invaluable in shaping the course of my research. I am grateful to Dinesh for accepting the role of co-advisor, contributing his expertise in optimization theory, and offering valuable perspectives that significantly enriched the quality of this work.

I express my gratitude to the PhD evaluation committee, comprising Professor Sebastian Engell, Dr. Elvira Marie Bergheim, and Associate Professor Idelfonso Nogueira, for dedicating their time to review and provide valuable feedback on this thesis. It is indeed a privilege to undergo evaluation by such a distinguished committee.

I am also sincerely thankful for the opportunity to collaborate with Dr. Edmary Altamiranda from AkerBP in the SUBPRO Innovation Project. The resources and practical aspects knowledge provided by Aker BP have significantly enriched my research experience, allowing me to delve into complex issues and explore innovative and pragmatic solutions.

I also extend my gratitude to all those with whom I have engaged in discussions, who have both directly and indirectly served as a wellspring of inspiration throughout my academic and professional journey since 2009. My sincere appreciation goes to Professor Bambang Riyanto Trilaksono, Professor Masayuki Fujita, Professor Siep Weiland, the late Professor Yudi S. Gondokaryono, Associate Professor Takeshi Hatanaka, Associate

Professor Leyla Ozkan, Associate Professor Mircea Lazar, Assistant Professor Azwirman Gusrialdi, Dr. Alejandro Marquez-Ruiz, Harwiyono, the late Liliek Eko Budoyo, Hartono, Arliansyah Abdul Gani, Achmad Falach, Iskandar Zulkarnain, Zulfian Arun, Irfansyah, the late Aznaldi Augustia, and many other colleagues whose insights and support have been instrumental in shaping my academic and professional endeavors.

I am also grateful for the opportunity to co-supervise master students: Kristian Odegård and Vegard Aas from NTNU. It was great to participate in their academic development.

I gratefully acknowledge the financial support from SFI SUBPRO, which is financed by the Research Council of Norway, major oil and gas industry partners and NTNU.

I extend my thanks to the vibrant community within the Process Systems Engineering group. Special acknowledgments to Allyne, Jose, Halvor, Rafael, Anna, Leonardo, Lucas B., Lucas C., Saket, Cristina, Zawadi, Fabienne, David, Simen, Rizwan, the late Sandeep, Andrea, Evren, Robert, Fernando, Carol, Peter, and other NTNU colleagues. Our engaging discussions, enjoyable happy hours, and memorable travels have significantly enriched my academic journey.

The last but the most important, I want to express my deepest gratitude to my parents; Revizal and Indrayani, my family and extended family. To my wonderful wife, Shintami, thank you for your unwavering love, support, and boundless patience. To my precious daughter, Thavisya, who became the most important 'publication' during my PhD journey, thank you for gracing our lives with your presence, laughter, and smiles. This thesis is dedicated to the memory of my late daughter, Mikayla, and to my living guardian angel, my son, Thariq. Thariq, your presence is the greatest gift in my life. You might never understand that one of my driving force to pursue PhD is to inspire and make you proud!

Risvan Dirza
Trondheim, June 2024

Contents

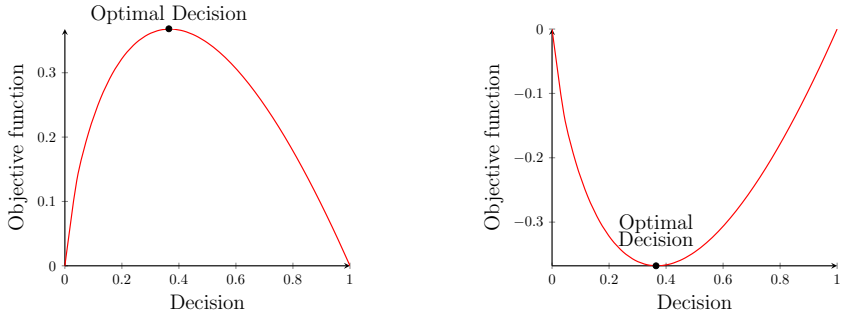
Chapter 1

Introduction

1.1 Optimization - Brief Theory and Oil & Gas Practice

1.1.1 Production Optimization

In production system operation, decisions span organizational levels and timescales, aiming to optimize performance by determining optimal decisions to maximize (for concave) or minimize (for convex) a well-defined scalar objective function consisting of product value and production costs (refer to Figure 1.1). Gathering pertinent information, such as physical properties and measurements, is crucial, and accounting for potential disturbances that can significantly impact performance is essential for effective optimization.



(a) Concave objective function

(b) Convex objective function

Figure 1.1: Relation between objective function and decision

1.1.2 Real-time Optimization

Real-time optimization (RTO) employs mathematical techniques to automate decision-making in determining optimal solutions, leveraging continuous analysis of real-time production data. The approach, extensively defined in [1], enables the identification of optimal decisions. As outlined in [2], an RTO system consists of four components, illustrated in Figure 1.2 where the grey box denotes the production system, and the white boxes symbolize the methods designed to constitute the RTO system, as follows:

- *Data acquisition and validation*: Data collected from the production system are validated through data reconciliation and pre-processing, accounting for physical definitions. In this thesis, we assume that this block will conduct a perfect data pre-processing to minimize noise or data error until an acceptable composition.
- *Parameter and State Estimation, and Model Update*: Using the validated data, parameters and/or states are updated through local estimators. These updates are incorporated into the model.
- *Steady-state optimization*: Given the complexity and scale of the problem, achieving an optimal solution under steady-state conditions is currently the primary focus.
- *Optimizer Command Conditioning*: A post-optimality analysis validates the proposed optimal solution, addressing potential numerical solver issues as a root cause necessitating this assessment.

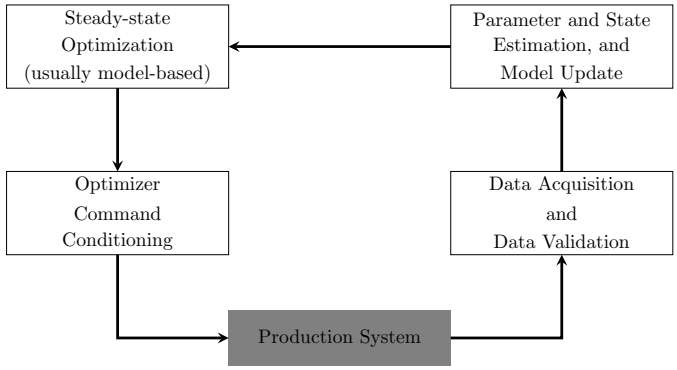


Figure 1.2: The components in a typical RTO system.

Figure 1.2 represents just one way to view an RTO; alternatively, it can be seen as an integrated system for optimization and parameter estimation [3]. A prominent approach adopting this perspective is Modifier Adaptation, first introduced in [4]. This approach addresses both parametric and structural plant-model mismatch by iteratively adjusting cost and constraint functions [5]. Furthermore, extensive research on this approach has continued over the past two decades, as evidenced by studies such as [6, 7] among others.

RTO involves diverse methods varying in complexity and flexibility, operating across different time scales (refer to Figure 1.3). The grey box denotes the production system, and the white boxes symbolize potential methods for development and implementation. These timescales categorize RTO into the following distinct groups:

- *Embedded Optimization (also known as feedback-optimizing control)*: This optimization framework employs a control structure to handle disturbances and attain (near-)optimal production performance on fast timescales. The self-optimizing control ensures an immediate response to disturbances by adjusting the process to the (near-)optimal region and maintaining a self-optimizing variable at a constant setpoint [8]. Assuming no measurement noise, the "ideal" self-optimizing variable corresponds to gradients [9], representing the cost function gradient for unconstrained or

the Lagrange function gradient for the constrained case. This framework utilizes model-based gradient estimation.

- *Model-based Optimization*: This popular optimization framework utilizes models for determining optimal decisions in medium timescales. Accurate decisions require model updates, yet a significant challenge is the steady-state wait time. In addressing this, [10, 11] proposed periodic parameter estimation for achieving optimality during transient periods.
- *Model-free Optimization*: This framework provides a cost-effective solution, especially for challenging or resource-intensive model development. Using input and measurement sets, it achieves (near-)optimal performance by estimating and controlling gradients to zero, satisfying the optimality condition without explicit reliance on a model [12]. However, gradient estimation can be time-consuming, making it more suitable for execution in slower timescales.
- *Large-scale Optimization*: This framework addresses large problems by decomposing them into manageable sub-problems/subsystems, coordinating subsystems in (*very*-) slow timescales, facilitating (*minimal*-) information exchange, and enhancing decision-making. For a comprehensive explanation of decomposing large problems into smaller subproblems, refer to [13].

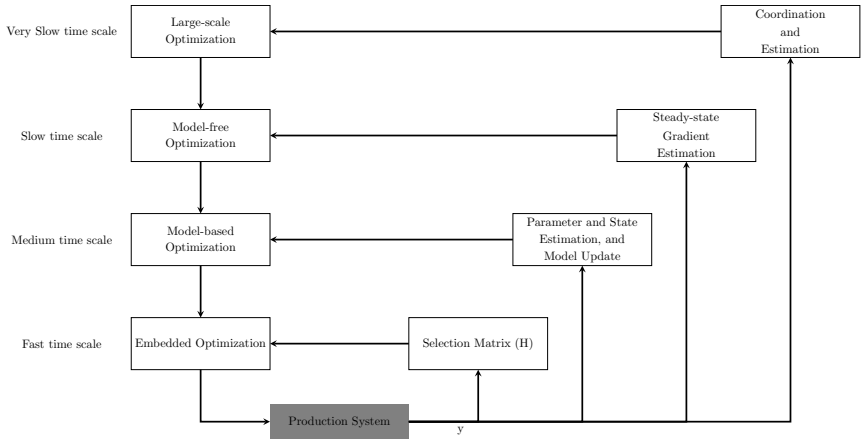


Figure 1.3: The structure of RTO based on its time-scale.

The slower timescale of large-scale optimization compared to model-free optimization can prompt investigation. This holds true in scenarios where large-scale optimization supports parameters in model-free optimization. Without model-free optimization, large-scale optimization can directly position above model-based optimization. The component arrangement should align with optimization problem specifics, ensuring efficiency.

With the advancements of data mining, processing, and analytics, there is a growing rationale for integrating machine learning to enhance RTO performance in parameter estimation, model development, and optimization algorithms. However, it is vital to emphasize that machine learning tools typically necessitate substantial and meaningful datasets for

optimal outcomes. Both the quantity and quality of available data are pivotal in determining the potential benefits of integrating machine learning into a system. Notably, machine learning may not be the most fitting approach for slow systems, such as process systems, raising intriguing avenues for further discourse and exploration.

1.1.3 Oil & Gas Production & Processing Optimization

Numerous researchers have extensively reviewed oil and gas production optimization, as demonstrated in studies like [1, 14]. However, pragmatic and commercial considerations often prompt simplification of the optimization problem by neglecting the influences of surface and other subsurface, risking suboptimal performance. Figure 1.4 illustrates *limitations* imposed by various commercial software utilized by different organization, hindering attainment of system-wide optimal performance. For example, organizations utilizing OLGA for multiphase flow simulation tend to assume simplified onshore process facilities parameters as constant values due to practicality and model accessibility.

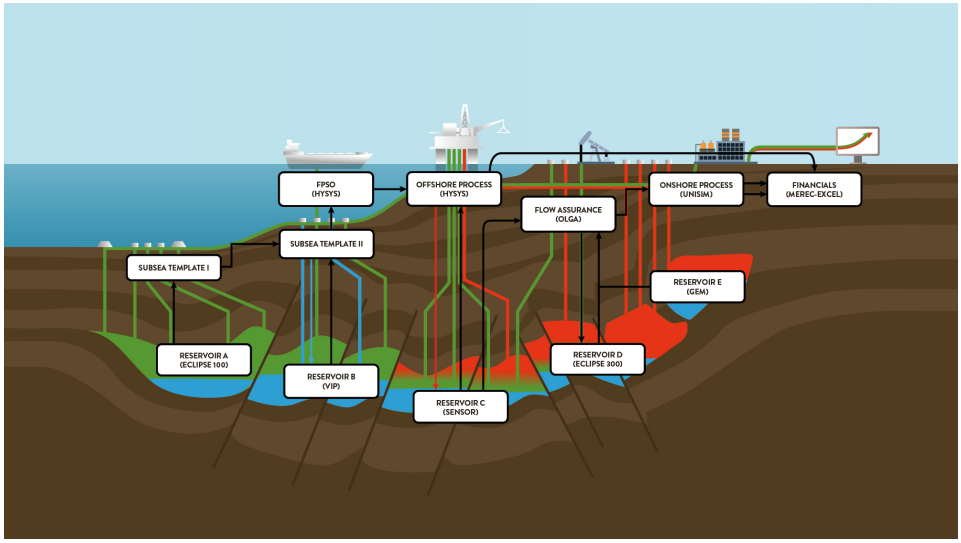


Figure 1.4: Variability in software adoption across organizations in oil and gas industry. Common commercial software, i.e., HYSYS, UNISIM, OLGA, VIP, used in each process system is written in parentheses.

Typically, a comprehensive production system comprises the reservoir system, well system, and processing system, each having its distinct models. The reservoir system, characterized by relatively slow dynamics in contrast to the well and processing systems, exerts a less impact on short-term optimizations. The well model undergoes regular updates to determine critical parameters like gas-oil ratio, water cut, and production rates, typically through well-testing procedures. Despite extensive research on automated well testing, manual testing remains the predominant practice.

The processing facilities model is presented as constraints on oil, gas, and water processing within a simplified production optimization framework. In practice, the capacity of each component cannot be investigated in isolation, necessitating an integrated optimiza-

tion approach encompassing both the well and processing facility systems. This integrated system is conceptualized as a serial-coupled production system. Additionally, a large-scale well network system can be viewed as a parallel coupled system.

In pursuit of the thesis objective, the author contends that realizing "true" (system-wide) optimization in oil and gas systems demands *integration of crucial elements within a coordinated RTO system for large-scale systems*. This holistic approach ensures optimal performance across the entire production and processing system.

1.1.4 RTO Components in Oil & Gas Practice

Data Acquisition and Validation: To effectively operate production systems, deploying suitable instrumentation for sensing and measurements is essential. The required instrumentation varies based on factors like system age, plant location, and company standards. Modern production systems commonly utilize advanced instrumentation to fulfill operational requirements, playing a vital role in data acquisition and control functions such as stabilizing the system through level and pressure control, avoiding surges, sustaining load balance among interconnected processing units, and slug control.

Parameter and State Estimation, and Model Update: The measurements obtained from production systems serve not only in control functions but also facilitate optimization through parameter, state, gradient, and coordination estimation. These estimations play a crucial role in updating the system's model when needed. Nevertheless, limitations may arise, especially with an inadequate set of available measurements, posing challenges to accurate estimations. However, harnessing extensive, long-term datasets has the potential to improve the accuracy of these estimations.

Planning and Optimization: Operating an oil and gas production system involves three key types of planning:

- *Strategic planning* integrates market dynamics, government policies, and company objectives, aligning production strategies with market demands and long-term goals.
- *Reservoir planning* entails formulating a cost-effective, long-term strategy for extracting hydrocarbons, while considering production constraints. It assesses reservoir characteristics, production limitations, and optimization factors to identify the optimal approach for maximizing hydrocarbon recovery.
- *Production planning* involves creating plans for production, injection, utility/energy consumption, and environmental compliance with green policies. These plans set target production rates (oil, gas, water), injection rates, and energy consumption over a specified period, typically ranging from a week to a month, following the field operator's policies.

Determining optimal set points involves solving a complex optimization problem considering an objective function and constraints. Achieving a global optimal solution of a complex production system is challenging, but making local optimal decisions based on the current operating point is a practical approach. In addition, expert judgment from field personnel familiar with the production system's behavior is essential. Although various numerical solvers address optimization problems, numerical issues may pose limitations. While RTO implementation in the oil and gas sector is not yet widespread, but its potential benefits are being explored. Therefore, *integrating operators and expert in the implementation of RTO is crucial for effective practical implementation*.

Command Conditioning: A highly complex oil and gas production system, involving uncertainties and measurement noise, requires discussions for the recommended optimal setpoint in morning meetings to evaluate their suitability amid current conditions. Numerical issues and factors like updated conditions, parameter changes, or assumptions can significantly affect the recommendation. Thus, acknowledging the dynamic nature of the production system is crucial for effective command conditioning, emphasizing the importance of considering evolving conditions and uncertainties.

1.2 Research Questions

The implementation of RTO in oil and gas production faces challenges due to the complexity and scale of the process, along with limited measurements, numerous constraints and uncertain parameters. Accurate parameter estimation becomes difficult, potentially resulting in suboptimal performance. Despite ongoing efforts to overcome these challenges, successful real-world applications of RTO in this domain are limited [15]. To enhance applicability, a practical strategy involves starting with a small scope that address specific sub-problems, and is coordinated to achieve system-wide optimal operation. However, technical challenges such as numerical robustness and computational issues may even hinder widespread adoption of RTO in industrial processes, leading engineers to rely on simpler feedback controllers for optimal operation. Given these realities, these factors collectively raise the main research question that this thesis aims to address as follows:

Main Research Question: *What strategies can be employed to optimize the operation of a complex, large-scale oil production system with diverse sub-processes, varying timescales, and potential numerous constraints, utilizing simple tools like PID controllers, selectors, and small-scale solvers (if necessary)?*

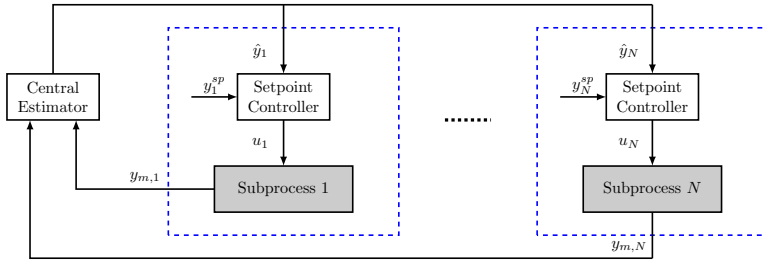
- The main research question stated above serves as the underlying theme of this thesis that guide the exploration and analysis conducted throughout all chapters.

To ensure a coherent information flow, this thesis breaks down the main research question into sub-research questions addressed across three distinct parts.

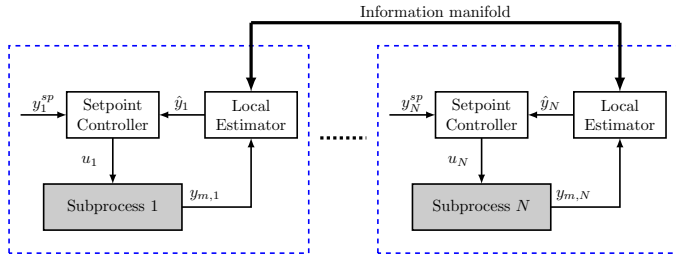
The first two parts discuss the schemes of *distributed feedback-optimizing control (DFoC)* and *distributed feedback-optimizing system (DFoS)*, explored further in this thesis. In Figure 1.5(a), DFoC optimizes individual sub-processes via decentralized control within a large-scale system, while a centralized estimator handles parameter and disturbance estimation. In this scheme, the term *distributed* specifically pertains to the control problem at hand. In Figure 1.5(b), DFoS coordinates interconnected sub-processes for system-wide optimization, integrating DFoC in each sub-process with local estimation and inter-subsystem coordination. The term *distributed* pertains to control and estimation.

1.2.1 Part I: Distributed Feedback-optimizing Control (For Small-scale System)

In complex hydrocarbon production systems, the presence of constraints, including both input and output constraints, can be extensive. They play a crucial role in maintaining safe and efficient system operations. It is important to note that under optimal operating conditions, the status of these constraints, whether active or inactive, can vary depending on disturbances values. Consequently, the first sub-question that arises is:



(a) Distributed Control



(b) Distributed System

Figure 1.5: Graphical illustration showcasing the concepts of *DFoC* and *DFoS*

Research Question 1: *How can we effectively utilize simple tools such as PID controllers, selectors and small-scale solvers (if necessary) to achieve optimal operation while effectively managing numerous constraints?*

- This question is addressed in Chapter 2 by suggesting Primal-Dual framework.

In oil and gas production, managing safety and environmental constraints is critical. The Primal-Dual control structure assumes soft constraints and controls them on a slower timescale, while enabling automatic active constraint switching. This leads to the second sub-question:

Research Question 2: *What strategy should we apply to minimize critical constraint violations in primal-dual framework (that sill allowing automatic constraint switching)?*

- This question is addressed in Chapter 3 by incorporating override constraint controller and *auxiliary* constraint.

1.2.2 Part II: Distributed Feedback-optimizing System (For Large-scale System)

Developing a RTO approach is practical to optimize small-scale systems like a single unit operation, employing both numerical solver-based and feedback-based methods, as described in Chapters 2-3. However, when dealing with complex oil and gas production systems with interconnected unit operations, effective coordination is vital for overall optimization. The *dFoS* is proposed to address this challenge, achieving system-wide optimization, handling sub-process failures, and allowing time scale separation among different units, enhancing efficiency and mitigating suboptimal performance issues. Typically, *dFoS*

can be categorized into dual and primal-based schemes that are discussed in Part II.1 and II.2, respectively.

Part II.1: Dual-based Distributed Feedback-optimizing System

Considering the need for *simple tools, preventing divergence, allowing different time scale separations among sub-processes, avoiding numerical robustness issues, and minimizing information sharing,* the third sub-question is stated as follows:

Research Question 3: *What strategy can be used to drive the entire system to achieve system-wide optimal operation, while satisfying the above requirements?*

- This question is addressed in Chapter 4 by proposing dual-based *dFoS* with and without a solution predictor, and is experimentally validated in Chapter 5.

Similar to the Primal-Dual control structure discussed earlier, the dual-based *dFoS* introduced in Chapter 4, operating within a system-wide framework, control constraints on a slower timescale while facilitating automatic active constraint switching. Addressing the need of minimizing constraint violations, particularly in critical aspects of oil and gas production operations, this gives rise to the fourth sub-question:

Research Question 4: *How can we improve constraint handling in a dual-based* dFoS *by that minimizes constraint violations? Additionally, how can we develop a systematic approach for pairing a MV with a critical constraint?*

- This question is addressed in Chapter 6 by also suggesting expected disturbance-based systematic pairing for override constraint control, and is experimentally validated in Chapter 9.

In cases involving parallel systems like gas-lift oil production optimization, multiple MVs can be concurrently utilized to control critical constraints. This leads the fifth sub-question:

Research Question 5: *How can we maximize the potential of multiple MVs (multi-input) for Override Constraint Control within the dual-based* dFoS *scheme?*

- This question is addressed in Chapter 7, where a comprehensive investigation on several potential procedures in constructing the selection matrix is conducted.

Part II.2: Primal-based Distributed Feedback-optimizing System

In this part, we investigates the potential of primal-based *dFoS*. One of the potential is the ability to satisfy constraint better than the dual-based *dFoS*. Thus, considering the requirements of using *simple tools, preventing divergence, allowing different time scale separations among sub-processes, avoiding numerical robustness issues, minimizing information sharing, facilitating better constraint handling to reduce back-off for critical constraints, and being practically less problematic for large-scale systems*, the sixth sub-question is formulated as follows:

Research Question 6: *What strategy can be used to drive the entire system to achieve system-wide optimal operation, while satisfying the above requirements?*

- This question is addressed in Chapter 8 by proposing primal-based *dFoS* with compensator, and is experimentally validated in Chapter 9.

1.2.3 Part III: Addressing practical issues

In this part, this thesis addresses several practical issues encountered in real-world scenarios, which are immediately identified in the corresponding research questions.

Primal-based *dFoS* offers a potential advantage; that is able to eliminate the need for a coordinator. This prevents partial issues among participating companies in a shared common resource. This leads to the seventh sub-question as follows:

Research Question 7: *What strategy can be used to drive the entire system to achieve system-wide optimal operation without centralized coordinator?*

- This question is addressed in Chapter 10 by proposing primal-based *dFoS* scheme as graph network.

Research Question 8: *How can we drive a recirculated gas-lifted subsea oil production system with limited measurements to achieve (near-) optimal performance?*

- This question is addressed in Chapter 11 by suggesting self-optimizing control.

1.3 Thesis Outline

This PhD study aimed to address the outlined research questions through chapters based on articles authored by the candidate, excluding the introductory (Chapter 1) and concluding (Chapter 12) chapters. Refer to Section 1.4 for a full list of the candidate's publications. Chapters 2 to 11 address research questions 1 to 8, while Chapter 9 focuses on experimentally validating the proposed approaches in Chapters 4, 6, and 8.

1.4 List of Publications

In four years, the PhD candidate has authored several peer-reviewed papers published in esteemed international conference proceedings, and three journal papers. Furthermore, the candidate has had the opportunity to showcase his research through numerous oral and poster presentations at various occasions.

1.4.1 Peer-reviewed Conference Papers

1. R. Dirza and S. Skogestad. *Handling Interactive Systems in Primal-Dual Feedback-optimizing Control.* In *Proceedings of the 3rd joint conference of Foundations of Computer Aided Process Operations and Chemical Process Control (FOCAPO/CPC)*, San Antonio, Texas, 2023 - **Chapter** 2.
2. R. Dirza, S. Skogestad, and D. Krishnamoorthy. *Optimal Resource Allocation using Distributed Feedback-based Real-time Optimization.* (IFAC Symposium on Advanced Control of Chemical Processes (ADCHEM), Venice, Italy). *IFAC-PapersOnline*, 54(3), pp. 706-711, 2021 (Keynote Paper) - **Chapter** 4.
3. R. Dirza, D. Krishnamoorthy, and S. Skogestad. *Primal-dual Feedback-optimizing Control with Direct Constraint Control for Oil Production.* (14th Symposium of Process Systems Engineering (PSE 2021+), Kyoto, Japan). *Computer-Aided Chemical Engineering*, 49, pp 1153-1158, 2022 - **Chapter** 6.
4. R. Dirza and S. Skogestad. *Systematic Pairing Selection for Economic-oriented Constraint Control.* (32nd European Symposium of Computer-Aided Process Engineering (ESCAPE 32), Toulose, France). *Computer-Aided Chemical Engineering*, 51, pp 1249-1254, 2022 - **Chapter** 6.

5. R. Dirza and S. Skogestad. *Online Feedback-based Optimization with Multi-input Direct Constraint Control.* (IFAC Symposium on Dynamics and Control of Process Systems (DYCOPS), Busan, Republic of Korea). *IFAC-PapersOnline*, 55(7), pp. 149-154, 2022 - **Chapter** 7.
6. R. Dirza, Md. Rizwan, S. Skogestad and D. Krishnamoorthy. *Real-time Optimal Resource Allocation using Online Primal Decomposition.* (IFAC Symposium on Control, Optimization, and Automation in Mining, Mineral and Metal Processing (MMM), Montreal, Canada). IFAC-PapersOnline, 55(21), pp. 31-36, 2022. - **Chapter** 8.
7. V. Aas, R. Dirza, S. Skogestad, and D. Krishnamoorthy. *A Comparative Study of Distributed Feedback-optimizing Control Strategies.* (33rd European Symposium of Computer-Aided Process Engineering (ESCAPE 33), Athens, Greece). *Computer-Aided Chemical Engineering*, 2023 - **Chapter** 9.
8. R. Dirza, S. Skogestad and D. Krishnamoorthy. *Real-Time Optimal Resource Allocation and Constraint Negotiation Applied to A Subsea Oil Production Network. SPE Annual Technical Conference and Exhibition (ATCE)*, Dubai, United Arab Emirates, 2021. - **Chapter** 10.
9. R. Dirza, E. Altamiranda and S. Skogestad. *Self-Optimizing Control for Recirculated Gas lifted Subsea Oil Well Production.* (IFAC Symposium on Advanced Control of Chemical Processes (ADCHEM), Toronto, Canada). *IFAC-PapersOnline*, 2024 (to appear) - **Chapter** 11.

1.4.2 Journal Papers

1. R. Dirza and S. Skogestad. *Primal-Dual Feedback-optimizing Control with Override for Real-time Optimization. Journal of Process Control.* 138, pp 103208, 2024 - **Chapter** 3.
2. R. Dirza, J.A. Matias, S. Skogestad, and D. Krishnamoorthy. *Experimental Validation of Distributed Feedback-based Real-Time Optimization in a Gas-lifted Oil Well Rig. Control Engineering Practice.* 126, pp 105253, 2022. - **Chapter** 5.
3. R. Dirza, V. Aas, S. Skogestad, and D. Krishnamoorthy. *A Comparative Study of Distributed Feedback-optimizing Control Schemes: An Experimental Validation. IEEE Transactions on Control Systems Technology*, 2024 (under review) - **Chapter** 9.

1.4.3 Additional Presentations (invited, or with abstract-only)

1. R. Dirza, S. Skogestad, and D. Krishnamoorthy, *Real-Time Optimal Resource Allocation Using Transient Measurements Based on Cooperative Game. Computer Aided Process Engineering (CAPE) Forum*, Copenhagen, Denmark, Oct 2020.
2. R. Dirza, S. Skogestad, and D. Krishnamoorthy, *Optimal Resource Allocation in a Subsea Oil Production Network Using Distributed Feedback-Based RTO. American Institute of Chemical Engineers (AIChE) Annual Meeting*, Boston, Massachusetts, USA, Nov 2021.
3. R. Dirza and S. Skogestad, *Generalized Primal-dual Feedback-optimizing Control with Direct Constraint Control. The* 23^{rd} *Nordic Process Control Workshop*, Luleå, Sweden, Mar 2022.

4. R. Dirza and S. Skogestad, *Dual-based Distributed Feeback-optimizing Control with Override. INTPART Brazil-Norway Subsea Operation Consortium Workshop*, Rio de Janeiro, Brazil, Nov 2022.
5. R. Dirza, S. Skogestad, and D. Krishnamoorthy, *Optimal Resource Allocation in a Subsea Oil Production Network Using Distributed Feedback-Optimizing Control Based on Primal Decomposition. American Institute of Chemical Engineers (AIChE) Annual Meeting*, Phoenix, USA, Nov 2022.
6. R. Dirza, J.A. Matias, S. Skogestad, and D. Krishnamoorthy, *Experimental Validation of Distributed Feedback-based Real-Time Optimization in a Gas-lifted Oil Well Rig. IFAC World Congress*, Yokohama, Japan, Jul 2023.
7. R. Dirza, K. Ødegård, E. Altamiranda and S. Skogestad, *Self-Optimizing Control for Recirculated Gas lifted Subsea Oil Well Production. The* 24^{th} *Nordic Process Control Workshop*, Trondheim, Norway, Aug 2023.

Part I: Distributed Feedback-optimizing Control (For Small-scale System)

Chapter 2

From Real Time Optimization to Primal Dual Framework

To be able to achieve optimal operation while effectively managing numerous constraints using simple tools such as Proportional–Integral–Derivative (PID) controllers, selectors and small-scale solver (if necessary), this chapter suggests and describe how Primal-Dual framework can be an effective solution for feedback-optimizing control. This chapter is the extended version of the work in [16].

2.1 Optimal Production Operation - State of the Art

As mentioned in the introduction, this thesis investigates the optimal production operation. But what does optimal production operation mean? This section will give an overview of optimal production operation and current practice in implementing it.

2.1.1 Control Structure of Optimal Production Operation

Each production system aims to optimize economic performance by maximizing product revenue and minimizing operating costs through process control. The control structure is often divided into multiple layers with varying time scales of execution, as depicted in Figure 2.1 (see [8, 17] to name a few). This hierarchy of the control structure consists of the following layers:

- *Scheduling* layer determines production timing, quantity, and location, often using *not* automated processes on a weekly time scale. It relies on simple linear models and shares similarities with asset and reservoir management layers in oil and gas production optimization.
- *Site-wide optimization* layer defines optimal operation setpoints for all units in the entire system on a daily time scale. It oversees multiple *local optimization* layers that utilize either simple linear or steady-state non-linear models based on complexity. In oil and gas production optimization, it is similar to a system-wide RTO layer.
- *Local optimization* layer determines optimal setpoints for an operation unit within a time scale of hours that utilizes steady-state or dynamic (non-) linear models based on complexity. In oil and gas production optimization, it resembles the local RTO.

- *Supervisory control* layer manages process control that ensures setpoint tracking from the layer above and satisfying operational constraints. It operates on a minute time scale and can be a Model Predictive Control (MPC) or a set of PID controllers, depending on the complexity of the system, such as in oil and gas production optimization.
- *Regulatory control* layer stabilizes process operations, typically within seconds, that utilizes a PID controller as the preferred tool in practical applications.

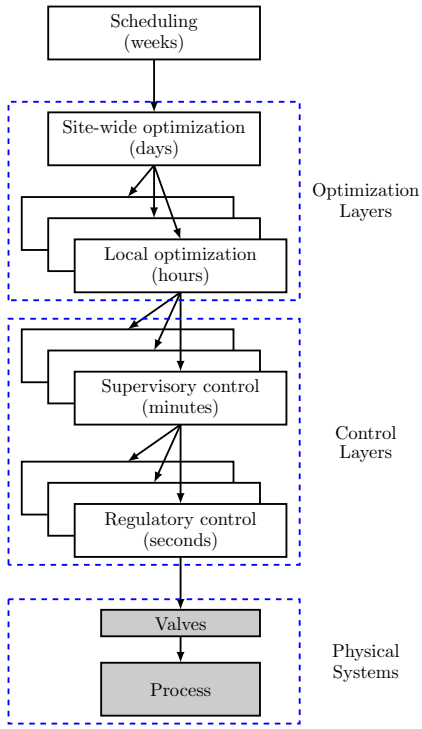


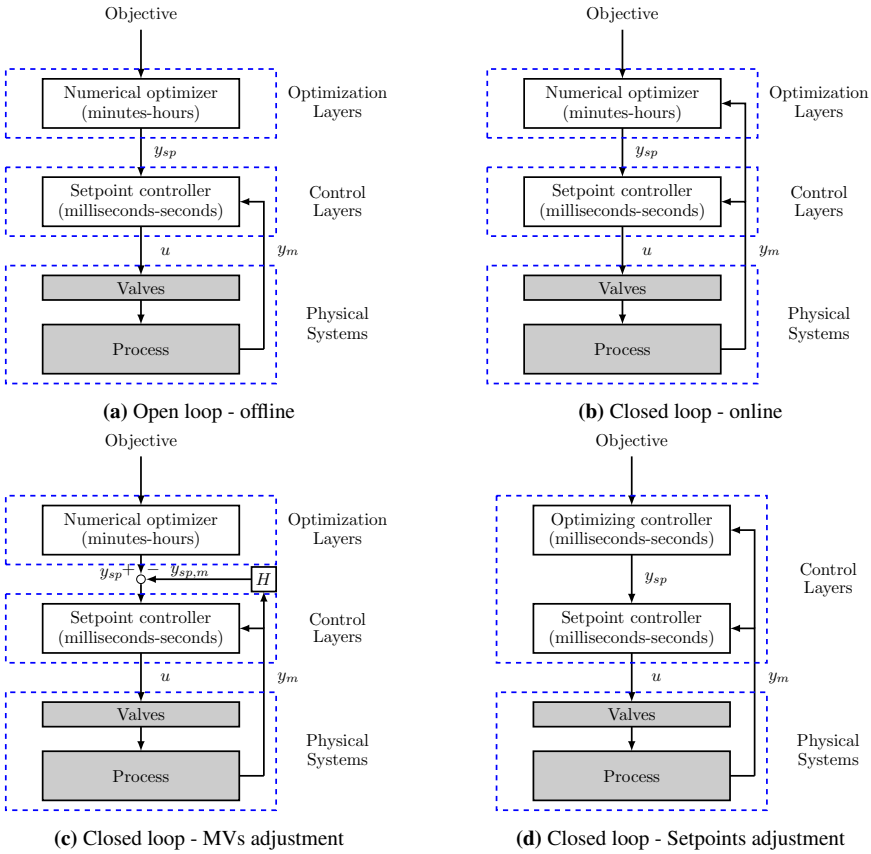
Figure 2.1: Typical control hierarchy

The layers are inter-connected through controlled variables. For example, the *site-wide optimization* layer calculates an optimal resource distribution, r_{sp}, for an operational unit, considering the overall production operation and current disturbances $\mathbf{d}$. The *local optimization* layer, utilizing the given resource r, determines an optimal setpoint c_{sp} for the controlled variable c under the current disturbances $\mathbf{d}$. Subsequently, the *supervisory control* layer strives to maintain the measured controlled variable c_m at the specified setpoint c_{sp} and the given resource r at the designated setpoint r_{sp}, all while adhering to operational constraints. This process persists until the *local optimization* layer provides new optimal setpoints. We typically assume a time scale separation between the layers, implying that adjustments to the setpoints are immediately effective.

Alternatively, an integrated optimization method is proposed, combining layers into a mixed integer dynamic optimization (MIDO) problem. This MIDO problem is then dis-

cretized into a mixed integer nonlinear programming (MINLP) problem using collocation point techniques. Despite its potential, this approach encounters computational complexity challenges, drawing attention from researchers in the field, as evident in studies such as [18, 19, 20].

To reduce the complexity, consider isolating the scheduling problem as an independent task. This is valid as scheduling solutions function on a slower timescale than both the *site-wide* and the *local optimization layers*. Thus, the control hierarchy in Figure 2.1 can be simplified into three layers: the *scheduling*, *optimization*, and *control layers*.



(a) Open loop - offline

(b) Closed loop - online

(c) Closed loop - MVs adjustment

(d) Closed loop - Setpoints adjustment

Figure 2.2: Possible implementations of optimal operation

The *optimization layers*, also known to as *RTO*, are pivotal for optimal real-time performance, comprising the *site-wide* and *local optimization layers*. This involves solving Nonlinear Programming (NLP) problems, aiming to optimize system performance by considering various objectives and constraints. To reduce computational complexity for practical implementation, dynamic RTO or economic MPC are excluded, justifying the formulation of the RTO as a steady-state optimization problem. On the other hand, the *control layer* includes the *supervisory* and *regulatory control layers*, responsible for track-

ing optimal setpoints from the optimization layer and ensuring process operation stability.

This chapter aims to explore various structures for the *optimization layer*, including the Primal-Dual framework. Figure 2.2 illustrates four potential approaches for optimizing production operations (Figure 2.2(a), 2.2(c), and 2.2(d) are adopted from [8]). The four approaches are:

- *Open-loop* implementation (excluded in this thesis) computes optimal setpoints *off-line*, lacking feedback. Thus, disturbances cause deviations from optimal operation.
- *Closed-loop* implementation computes optimal setpoints *online*, incorporating feedback to adjust them in response to disturbances. Section 2.1.3, 2.1.4, 2.2.1, and 2.2.2 are the modification of this implementation.
- *Closed-loop* implementation with a control layer *adjusts the manipulated variables used for controlling the controlled variables to the setpoint given by the optimization layer* in response to disturbances. This implementation can be seen in section 2.1.5.
- *Closed-loop* implementation with optimizing controller *adjusts the setpoint of the manipulated variables* given by the optimizing-controller by providing feedback in response to disturbances. This implementation can be seen in section 2.3.1-2.3.2.

The *Process* block in Figure 2.2 corresponds to the process as seen by the optimization layer, potentially incorporating stabilizing controllers, and is not necessarily uncontrolled. In such instances, the optimization layer determines the setpoints for the controllers.

2.1.2 Plantwide Control Structure Design Procedure

In 2004, a systematic procedure was proposed for constructing an optimal plantwide control structure in order to attain (near-) optimal performance. For details, refer to [21]. A crucial element of this procedure emphasizes the important of active constraint control. In moderately linear systems, optimal performance is generally achieved at the intersection of multiple active constraints [22]. Varied disturbances lead to different configurations of active constraints, represented as distinct regions—an idea denoted as active constraint regions [23].

2.1.3 Traditional Steady-state RTO

The traditional steady-state RTO implementation, as illustrated in Figure 2.3, follows a two-step approach. Here is a brief summary of the process:

- Step 1: Steady-state parameter update

  The initial steps involves detecting steady-state conditions and preprocessing data to ascertain the plant's proximity to steady-state for initiating the RTO sequence. Techniques like statistical and trend analysis are employed to identify steady-state conditions. Once detected, the subsequent step entails parameter estimation using regression techniques to adjust model parameters based on current data. This process includes data reconciliation and model adaptation, reconciling measurement data with process models and updating the model accordingly. Rigorous screening of measurement data is conducted to identify and rectify errors before parameter estimation. Expert process knowledge is crucial for determining which model parameters require updating, as emphasized by [24].

- Step 2: Steady-state optimization

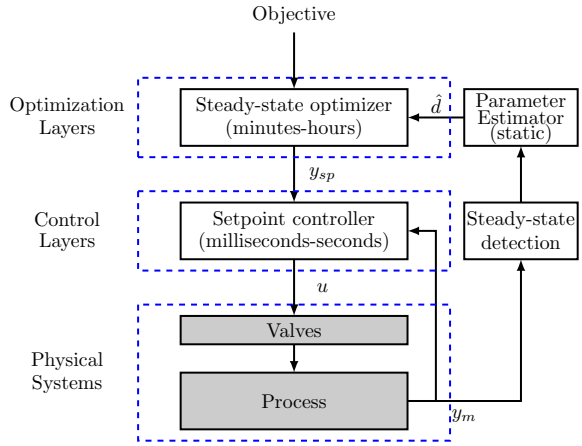


Figure 2.3: RTO with steady-state model adaptation and steady-state optimization

Utilizing an objective function, process constraints, and an updated model, mathematical optimization methods compute the optimal system setpoints. The objective function establishes the optimization goal, such as maximizing production and/or minimizing operating costs. Process constraints, including physical and operational limits, are considered to ensure setpoints meet system requirements. Techniques like linear or nonlinear programming are then applied to find the optimal solution.

Remark 2.1: Issues of Traditional Steady-state RTO

Commercial software often relies on statistical or heuristic methods, or a combination of both, to evaluate data stationarity within a fixed window. However, specifying tolerances without proper consideration of the data window length may erroneously accept transient data as steady-state, causing estimation errors and inaccuracies in the optimization routine. [25] illustrated this issue with real industrial data. Disturbed processes, with prolonged transient periods, pose challenges for model adaptation, especially when settling times are long. In such cases, limited access to steady-state measurements impedes frequent parameter updates, resulting in suboptimal operation until parameters can be updated again.

2.1.4 Steady-state RTO with Dynamic Model Adaptation

To address concerns outlined in Remark 2.1, [10, 11] proposed an RTO with steady-state optimizer and periodic dynamic model adaptation, depicted in Figure 2.4. In dynamic model adaptation, a dynamic model is used and its parameters are updated based on the available measurements, while in traditional RTO only a steady-state model is updated. Here is a brief summary of steady-state RTO with Dynamic Model Adaptation.

- Step 1: Steady-state parameter update

  The dynamic model's uncertain variables are periodically estimated using various methods, including recursive least squares, nonlinear Kalman filter [26] variants

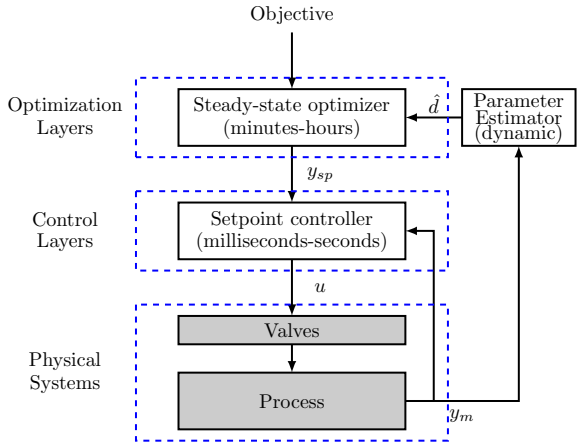


Figure 2.4: RTO with dynamic model adaptation and steady-state optimization

(e.g., EKF and UKF), and optimization-based approaches like the moving horizon estimator (MHE). Figure 2.4 is adapted from [10], suggesting that in specific scenarios, differential state estimation may not be necessary. However, in most cases, the steady-state estimator requires both disturbance ($\hat{d}$) and differential state ($\hat{x}$) estimation as part of its input.

- Step 2: Steady-state optimization

  This step corresponds precisely to Step 2 of the traditional steady-state RTO.

Throughout this thesis, we interchangeably use the terms "Hybrid RTO" and "Steady-state RTO with Dynamic Model Adaptation".

Remark 2.2: Issues of Steady-state RTO with Dynamic Model Adaptation

This approach lacks transparent constraint control (refer to Definition 2.1) and relies on a constraint model vulnerable to model mismatch. It contradicts the plantwide control structure design in Section 2.1.2. The procedure emphasizes prioritizing control of active constraints before process optimization.

Definition 2.1: Transparent Constraint Control

Transparent constraint control is a technique of constraint control that controls a *measured* constrained variable (not through a constraint model) using a simple feedback controller such as a PID controller to update manipulated variables.

2.1.5 Traditional Self-optimizing Control

In earlier sections, controlled variables play a crucial role in linking different layers within a control hierarchy. The process of selecting these variables raises a fundamental question [27]. Researchers have worked on the design of control structures and the primary selection of controlled variables [8, 9, 23, 28, 29, 30, 31]. Notably, [8] introduces the concept of self-optimizing control, depicted in Figure 2.2(c).

Definition 2.2: Traditional Self-optimizing Control

Traditional self-optimizing control is when we can achieve an acceptable loss with constant setpoint values for the controlled variables (without the need to reoptimize when disturbances occur)

Definition 2.3: Self-optmizing Controlled Variables

The controlled variables associated with self-optimizing control is also known as *self-optimizing controlled variables*.

As self-optimizing control is looking at minimizing the steady-state loss, we consider the following steady-state optimization problem.

$$\min_{\mathbf{u}} \quad J(\mathbf{u}, \mathbf{x}, \mathbf{d}) \tag{2.1a}$$

$$\text{s.t.} \quad \mathbf{f}(\mathbf{u}, \mathbf{x}, \mathbf{d}) = \mathbf{0}, \tag{2.1b}$$

$$\mathbf{g}(\mathbf{u}, \mathbf{x}, \mathbf{d}) \leq \mathbf{0} \tag{2.1c}$$

where $\mathbf{x} \in \mathbb{R}^{n_\mathbf{x}}$ are the state variables, $\mathbf{d} \in \mathbb{R}^{n_\mathbf{d}}$ are the disturbance variables, and $\mathbf{u} \in \mathbb{R}^{n_\mathbf{u}}$ are the decision/manipulated variables/degree of freedom/inputs. In addition, the label of $n_\mathbf{x}$, $n_\mathbf{d}$, and $n_\mathbf{u}$ are the number of states, disturbances and degree of freedom, respectively. The dynamic behaviour of the process is described by $\mathbf{f}$, and the operational constraints on the states, inputs and disturbances are imposed through $\mathbf{g}$.

To calculate self-optimizing controlled variables, traditional approaches often employ linear combinations of measurements, raising the question of optimal variable selection. Addressing this query involves identifying active constraint regions, tightly controlling the active constraints, and according to [8], a self-optimizing controlled variables should have the following properties;

1. The optimal value of the self-optimizing controlled variables should be insensitive to disturbances so that the setpoint error is small.
2. Select easily measurable and controllable variables for self-optimization to minimize implementation errors. Due to the difficulty in estimating the cost function gradient w.r.t. manipulated variables, this method excludes the gradient as self-optimizing controlled variables.
3. The gain from the manipulated variables to the self-optimizing controlled variable should be large. This corresponds to a flat optimum with respect to the controlled variable. This implies that the cost is insensitive to value of the controlled variable, and therefore, large gain enhances the ability to maintain optimal operation. In addition, the chosen manipulated variables should not be saturated (or far away from saturation).
4. The self-optimizing controlled variables should not be closely related in the case of several self-optimizing variables.

A concise summary of this approach can be found in [32, 33]. One interesting method is self-optimizing control (SOC) using the null space method.

Theorem 2.1: SOC using the null space method

Let $\mathbf{u_{DOF}}$ be the remaining degree of freedom, $\mathbf{y} \in \mathbb{R}^{n_\mathbf{y}}$ is the output, $\mathbf{G^y} \in \mathbb{R}^{n_\mathbf{y} \times n_\mathbf{u}}$ is the process gain matrices, $\mathbf{G_d^y} \in \mathbb{R}^{n_\mathbf{y} \times n_\mathbf{d}}$ is disturbance gain matrices, and assuming noise free measurements, $\mathbf{y} = \mathbf{G^y u_{DOF}} + \mathbf{G_d^y d}$. If there exist $n_\mathbf{y} \geq n_\mathbf{DOF} + n_\mathbf{d}$ independent measurements (independent here means that $\tilde{\mathbf{G}}^\mathbf{y} = \begin{bmatrix} \mathbf{G^y} & \mathbf{G_d^y} \end{bmatrix}$ has full low rank), then there exist $n_\mathbf{DOF}$ linear variable combinations $\mathbf{c} = \mathbf{Hy}$, $\mathbf{c} \in n_\mathbf{DOF}$, which are invariant to disturbances $\mathbf{d}$. $\mathbf{H}$ is selected such that $\mathbf{HF} = \mathbf{0}$ where

$$\mathbf{F} = \frac{\partial \mathbf{y}^\star}{\partial \mathbf{d}} \tag{2.2}$$

Controlling $c = \mathbf{Hy}$ to $c^{sp} = \mathbf{Hy}^\star$ (for unconstrained case, $c^{sp} = 0$) yields optimal operation with zero loss, that is the optimal value of $\mathbf{c}$ is independent of $\mathbf{d}$ [29].

Remark 2.3: Issues of Traditional SOC

The traditional SOC aims to minimize steady-state loss and attain near-optimal performance in the presence of expected disturbances, without necessitating re-optimization. Yet, relying solely on linear combinations of measurements does not guarantee optimal performance in uncertainties. Moreover, this approach assumes a constant set of active constraints.

2.2 Numerical Solver-based RTO with Transparent Constraint Controller

Developing a numerical solver-driven RTO with a transparent constraint controller addresses issue outlined in Remark 2.2. To overcome this challenge, two potential structures have been identified: the Reduced Gradient Hybrid RTO and the Dual-based Hybrid RTO, to be explored in subsequent sections. It is crucial to emphasize that a numerical solver-based RTO is recommended for optimizing highly complex processes, where pairing numerous PID controllers can be a daunting and sometimes impractical.

2.2.1 Reduced Gradient Hybrid RTO

The traditional SOC approach begins by identifying the regions where active constraints are present and ensuring tight control over these constraints as the primary priority. Once these constraints are controlled, remaining degrees of freedom are used to solve an equality problem. This involves finding a solution that eliminates the presence of the Lagrange multipliers [23, 34], where linear combinations of gradients are set to zero, converting it into an unconstrained steady-state optimization problem solvable that can be solved using an equation or numerical solver. Gradients are estimated using methods discussed in [35], with the following assumption:

Assumption 2.1: Sufficient differentiable conditions

Cost or objective function (2.1a) is sufficiently continuous and twice differentiable such that for any disturbance $\mathbf{d}$, cost or objective function (2.1a) has a (local-) minimum at $\mathbf{u} = \mathbf{u}^\star$. According to the KKT conditions (unconstrained case), the fol-

lowing then holds:

$$\nabla_{\mathbf{u}}\mathbf{J}\left(\mathbf{u}^{\star},\mathbf{x},\mathbf{d}\right)=\mathbf{0} \qquad \nabla_{\mathbf{uu}}\mathbf{J}\left(\mathbf{u}^{\star},\mathbf{x},\mathbf{d}\right)\geq\mathbf{0}$$

Considering complete KKT conditions (constrained case), Assumption 2.1 can be generalized as shown in Assumption 2.2

Assumption 2.2: Complete sufficient differentiable conditions

Cost or objective function (2.1a) is sufficiently continuous and differentiable, active constraint $\mathbf{g}_{\mathbb{A}}\left(\mathbf{u},\mathbf{x},\mathbf{d}\right)$ is sufficiently continuous and differentiable such that for any disturbance $\mathbf{d}$, there exists a (local-) minimum at $\mathbf{u}=\mathbf{u}^{\star}$, and let $\mathbf{N}\in\mathbb{R}^{n_{\mathbf{u}}\times\left(n_{\mathbf{u}}-n_{\mathbf{a}}\right)}$ be the nullspace of the active constraint gradients $\nabla_{\mathbf{u}}\mathbf{g}_{\mathbb{A}}\left(\mathbf{u},\mathbf{x},\mathbf{d}\right)$. According to the KKT conditions, the following then holds:

$$\mathbf{N}^{\top}\nabla_{\mathbf{u}}\mathbf{J}\left(\mathbf{u}^{\star},\mathbf{x},\mathbf{d}\right)=\mathbf{0} \qquad \mathbf{N}^{\top}\nabla_{\mathbf{uu}}\mathbf{J}\left(\mathbf{u}^{\star},\mathbf{x},\mathbf{d}\right)\geq\mathbf{0}$$

which is equivalent to

Let $\mathbf{u_{DOF}}$ be the remaining unconstrained degree of freedom, the following then holds:

$$\nabla_{\mathbf{u_{DOF}}}\mathbf{J}\left(\mathbf{u_{DOF}}^{\star},\mathbf{x},\mathbf{d}\right)=\mathbf{0} \qquad \nabla_{\mathbf{u_{DOF}u_{DOF}}}\mathbf{J}\left(\mathbf{u_{DOF}}^{\star},\mathbf{x},\mathbf{d}\right)\geq\mathbf{0}$$

which is also equivalent to

Cost or objective function (2.1a) is sufficiently continuous and differentiable, constraint function (2.1c) is sufficiently continuous and differentiable, and let $\boldsymbol{\lambda}\in\mathbb{R}^{n_{\mathbf{g}}}$ be the Lagrange multipliers of the constraint (2.1c). According to the KKT conditions, the following then holds:

$$\nabla_{\mathbf{u}}\boldsymbol{\mathcal{L}}\left(\mathbf{u}^{\star},\mathbf{x},\mathbf{d}\right)=\mathbf{0} \qquad \nabla_{\mathbf{uu}}\boldsymbol{\mathcal{L}}\left(\mathbf{u}^{\star},\mathbf{x},\mathbf{d}\right)\geq\mathbf{0}$$

This approach corresponds to the closed-loop implementation with an online numerical solver, as depicted in Figure 2.2(b). However, a more comprehensive block diagram for the Reduced Gradient Hybrid RTO is shown in Figure 2.5.

To describe reduced gradient approach more detail, consider the Lagrange function of problem (2.1) as follows.

$$\mathcal{L}\left(\mathbf{u},\mathbf{x},\mathbf{d},\boldsymbol{\lambda}\right)=J\left(\mathbf{u},\mathbf{x},\mathbf{d}\right)+\boldsymbol{\lambda}^{\top}\mathbf{g}\left(\mathbf{u},\mathbf{x},\mathbf{d}\right) \tag{2.3}$$

where $\boldsymbol{\lambda}\in\mathbb{R}^{n_{\mathbf{g}}}$ is the Lagrange multipliers associated with the inequality constraint (2.1c). Here, we substitute steady-state equality constraint (2.1b) in the cost function (2.1a).

The Karush-Kuhn-Tucker (KKT) conditions for stationary conditions is when

$$\nabla_{\mathbf{u}}\boldsymbol{\mathcal{L}}\left(\mathbf{u},\mathbf{x},\mathbf{d},\boldsymbol{\lambda}\right)=\mathbf{0}$$

Depending on the disturbances, different combinations of constraints may be active. Active constraints mean a set of constraints $\mathbf{g}_{\mathbb{A}}\left(\mathbf{u},\mathbf{x},\mathbf{d}\right)\subseteq\mathbf{g}\left(\mathbf{u},\mathbf{x},\mathbf{d}\right)$ that are optimal at its limit. Let $n_{\mathbf{g}_{\mathbb{A}}}\leq n_{\mathbf{g}}$ denote the number of active constraints $\mathbf{g}_{\mathbb{A}}\left(\mathbf{u},\mathbf{x},\mathbf{d}\right)$. The Karush-Kuhn-Tucker (KKT) conditions for complementary slackness condition states that, for the

active inequality constraints $\mathbf{g}_{\mathbb{A}}(\mathbf{u},\mathbf{x},\mathbf{d})=\mathbf{0}$, the corresponding Lagrange multipliers are positive $\boldsymbol{\lambda}_{\mathbb{A}}>\mathbf{0}$ and for the inactive inequality constraint $\mathbf{g}_{\mathbb{I}}(\mathbf{u},\mathbf{x},\mathbf{d})<\mathbf{0}$, the corresponding Lagrange multipliers are zero, $\boldsymbol{\lambda}_{\mathbb{I}}=\mathbf{0}$. Therefore, stationary conditions can be simplified as follows,

$$\nabla_{\mathbf{u}}J(\mathbf{u},\mathbf{x},\mathbf{d})=-\boldsymbol{\lambda}_{\mathbb{A}}^{\top}\nabla_{\mathbf{u}}\mathbf{g}_{\mathbb{A}}(\mathbf{u},\mathbf{x},\mathbf{d}) \tag{2.4}$$

[23] eliminates the Lagrange multipliers $\boldsymbol{\lambda}_{\mathbb{A}}$ by looking into the nullspace of the active constraint gradients $\nabla_{\mathbf{u}}\mathbf{g}_{\mathbb{A}}(\mathbf{u},\mathbf{x},\mathbf{d})$, which is basically based on the Frank-Wolfe's Reduced Gradient algorithm made by Abadie-Carpenter to handle nonlinear constraint [36]. This algorithm has been realized in the well-known software CONOPT.

The necessary conditions of optimality can be achieved by solving the following linear combination of the gradients estimation to constant setpoints of zero,

$$\mathbf{N}^{\top}\nabla_{\mathbf{u}}\mathbf{J}(\mathbf{u},\mathbf{x},\mathbf{d})=\mathbf{0} \tag{2.5}$$

where $\mathbf{N}\in\mathbb{R}^{n_{\mathbf{u}}\times(n_{\mathbf{u}}-n_{\mathbf{a}})}$, and $\mathbf{N}$ is defined as the nullspace of $\nabla_{\mathbf{u}}\mathbf{g}_{\mathbb{A}}(\mathbf{u},\mathbf{x},\mathbf{d})$ if $\mathbf{N}^{\top}\nabla_{\mathbf{u}}\mathbf{g}_{\mathbb{A}}(\mathbf{u},\mathbf{x},\mathbf{d})=\mathbf{0}$. We call $\mathbf{N}^{\top}\nabla_{\mathbf{u}}\mathbf{J}(\mathbf{u},\mathbf{x},\mathbf{d})$ the reduced gradient [37]. We assume linear independence constraint qualification (LICQ) hold to guarantee the existence of Lagrange multipliers.

Problem (2.5) can be solved numerically by formulating the following unconstrained steady-state optimization problem,

$$\min_{\mathbf{u}} \quad ||\mathbf{N}^{\top}\nabla_{\mathbf{u}}\mathbf{J}(\mathbf{u},\mathbf{x},\mathbf{d})|| \tag{2.6}$$

To be able to solve problem (2.6), the decision variable, that is primal variable $\mathbf{u}$, should exist. Therefore, the following assumptions must hold.

Assumption 2.3: The decision variables explicitly appear

Cost function of problem (2.1) is sufficiently continuous and the decision variables explicitly appear in the first derivative of the cost function. For example, at least, twice differentiable such that the decision variables explicitly appear in the gradient of cost function $\nabla_{\mathbf{u}}\mathbf{J}(\mathbf{u},\mathbf{x},\mathbf{d})$.

Assumption 2.4: Unique optimal decision variables

Problem (2.6) is guaranteed to have a unique solution within its searching space.

Theorem 2.2: Reduced Gradient HRTO

Assuming noise free measurements, Assumption 2.1, and Assumption 2.3 is valid. If there exist $n_{\mathbf{u}}\geq n_{\mathbf{a}}$, or equaivalently, $n_{\mathbf{DOF}}\geq 0$, and let $\mathbf{N}\in\mathbb{R}^{n_{\mathbf{u}}\times(n_{\mathbf{u}}-n_{\mathbf{a}})}$ be the nullspace of the active constraint gradients $\nabla_{\mathbf{u}}\mathbf{g}_{\mathbb{A}}(\mathbf{u},\mathbf{x},\mathbf{d})$ such that $\mathbf{N}^{\top}\nabla_{\mathbf{u}}\mathbf{g}_{\mathbb{A}}(\mathbf{u},\mathbf{x},\mathbf{d})=\mathbf{0}$, solving an equality problem or a steady-state optimization problem (2.6) yields optimal operation with zero loss, that satisfy the necessary conditions of optimality.

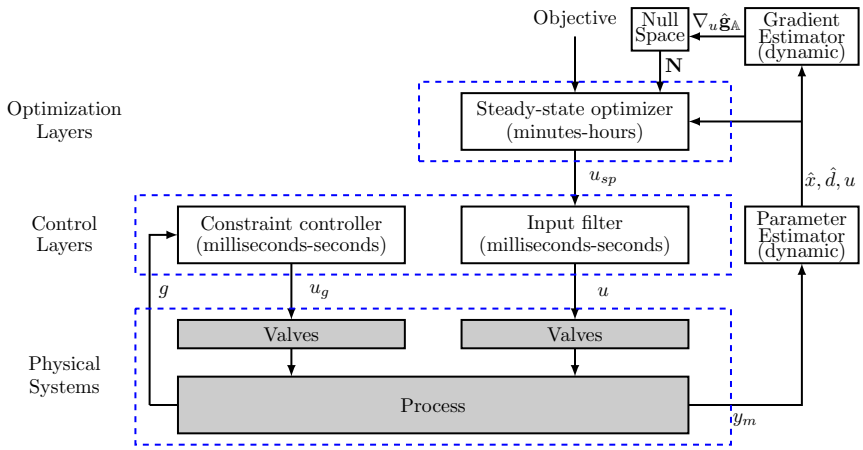


Figure 2.5: Reduced gradient hybrid RTO

In order to ensure a smooth (not too aggressive) implemented input, $\mathbf{u}$, we suggest to use a first-order input filter combined with input rate limiter as follows,

$$\mathbf{u}(k) = \mathbf{u}(k-1) + \mathbf{K_u}(\Delta \mathbf{u}(k)) \tag{2.7}$$

where $\Delta \mathbf{u}^{min} \leq \mathbf{K_u} \Delta \mathbf{u}(k) \leq \Delta \mathbf{u}^{max}$, $\Delta \mathbf{u}(k) = \mathbf{u}_{sp}(k) - \mathbf{u}(k-1)$, where $\mathbf{u}_{sp}$ is the solution from the steady-state solver, the diagonal matrix $\mathbf{K_u} < \mathbf{I}$ is the filter gain. Note that $\mathbf{K}_{u_i} = 1/(1 + \boldsymbol{\tau_{f_i}}/\Delta t)$, where τ_{f_i} is the filter time constant, and Δt is the sampling time. For the case with relatively mild disturbance, the input rate limiters may not be necessary. The input filter may be needed to achieve smooth changes of the inputs to the plant.

Also note that a direct implementation of the inputs (primal variables) into the process may result in instability (because the estimator might not immediately provide accurate state/parameter estimation), so in most cases, a lower layer controller, such as a first-order filter (or a setpoint controller), would need to be added.

If the solver can provide optimal setpoint of certain output (y^{sp}), it is possible to replace the input filter with a setpoint controller.

Remark 2.4: Issues of Reduced Gradient HRTO

The nullspace of active constraint gradients, denoted as $\mathbf{g}_{\mathbb{A}}(\mathbf{u}, \mathbf{x}, \mathbf{d})$, is determined by active constraints. However, disturbances can activate different constraints, requiring identification of all possible nullspaces for various combinations. This becomes impractical for processes with numerous constraints. Moreover, the steady-state optimizer may require significant time to provide the solution, which can be problematic as disturbances typically occurs in faster time scale. Furthermore, the presence of the optimizer may cause numerical robustness issues, and the need to ensure decision variable presence in the first derivation can limit applicability.

2.2.2 Dual-based Hybrid RTO

Briefly, the term of *dual* refers to the idea of hierarchical decomposition of the optimization problem where the constraint is controlled by dual variables. In many references we cited and this Chapter, we also use the term of *primal-dual*.

In Remark 2.3 and 2.4, both traditional SOC and reduced gradient HRTO require constructing a plantwide control structure within each active constraint region and determining switching strategies, collectively known as the *region-based approach*. These approaches assume that the a priori identification of active constraint regions, where active constraints are tightly controlled, focusing solely on the reduced unconstrained optimization problem. Changes in active constraint regions are typically managed using classical advanced control tools. This results in an exponential increase in the number of the structures relative to the number of constraints, specifically $2^{n_{\mathbf{g}}}$ for $n_{\mathbf{g}}$ constraints.

To address issues raised in Remark 2.3 and 2.4, inspired by [38], this study suggests transforming the constrained optimization problem into an unconstrained one using Lagrangian relaxation. Optimal operation is achieved asymptotically by controlling constraints through Lagrange multipliers/dual variables, complemented with selectors. The unconstrained optimization problem, dependent on dual variables, is solved in a cascading fashion to determine the optimal setpoint or physical manipulated variables/primal variables—a method commonly termed the *primal-dual approach*.

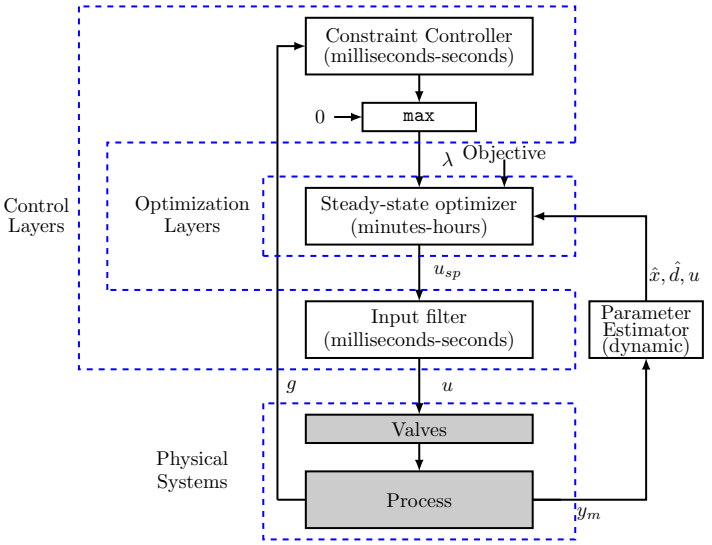


Figure 2.6: Dual hybrid RTO

The main idea involves *transparently* controlling the constraints in the upper layer by manipulating dual variables, termed *master controllers* in a cascade structure and *constraint controllers* based on their function. Each constraint is controlled by a simple single-loop I-controller, complemented with selectors for inequality constraints, in order to satisfy the complementary condition. When a constraint is active, the associated dual variable is positive and therefore the constraint is considered in the unconstrained optimization prob-

lem. Using the selector, the associated dual variable is zero if the constraint is inactive, and therefore the unconstrained optimization problem does not consider that constraint in the formulation. This approach corresponds to the closed-loop implementation with an online numerical solver, as depicted in Figure 2.2(b). However, a more comprehensive block diagram is shown in Figure 2.6.

Problem Re-formulation

Using Lagrangian relaxation, *constrained* steady-state optimization problem (2.1) can be reformulated as the following *unconstrained* steady-state optimization problem (see [38, 39, 40, 41, 42], and Fig. 2.6),

$$\min_{\mathbf{u}} \quad \boldsymbol{\mathcal{L}}\left(\mathbf{u}, \mathbf{d}, \boldsymbol{\lambda}\right) \tag{2.8}$$

where $\boldsymbol{\mathcal{L}}\left(\mathbf{u}, \mathbf{d}, \boldsymbol{\lambda}\right)$ is the Lagrange function as defined in Eq. (2.3), $\mathbf{u} \in \mathbf{R}^{n_{\mathbf{u}}}$ is the *primal variables*, and $\boldsymbol{\lambda} \in \mathbf{R}^{n_{\mathbf{g}}}$ is the *dual variables* associated with the (in-)equality constraints $\mathbf{g}$. Note that, for the sake of simplicity, we do not explicitly display the internal states $\mathbf{x}$ in the formulation.

Based on problem formulation (2.8), we want to achieve KKT optimality conditions as described in the Theorem 2.3.

Theorem 2.3: Karush-Khun-Tucker (KKT) Optimality Conditions

Suppose that the objective function $J\left(\mathbf{u}, \mathbf{d}\right)$ and constraint $\mathbf{g}\left(\mathbf{u}, \mathbf{d}\right)$ have subderivatives at point $\mathbf{u}^{\star}$. If $\mathbf{u}^{\star}$ is a local optimum and the optimization problem satisfies some regularity or *KKT conditions* (see below), then there exist constants $\boldsymbol{\lambda}$, called *KKT multipliers* or *Lagrange multipliers* or *dual variables* , such that the following conditions hold:

$$\nabla_{\mathbf{u}} \boldsymbol{\mathcal{L}}\left(\mathbf{u}, \mathbf{d}, \boldsymbol{\lambda}\right) = \mathbf{0} \tag{2.9a}$$
$$g_i(\mathbf{u}, \mathbf{d}) \leq 0, \ \forall i = 1, \ldots, n_{\mathbf{g}} \tag{2.9b}$$
$$\lambda_i \geq 0, \ \forall i = 1, \ldots, n_{\mathbf{g}} \tag{2.9c}$$
$$\lambda_i g_i(\mathbf{u}, \mathbf{d}) = 0, \ \forall i = 1, \ldots, n_{\mathbf{g}} \tag{2.9d}$$

where

$$\nabla_{\mathbf{u}} \boldsymbol{\mathcal{L}}\left(\mathbf{u}, \mathbf{d}, \boldsymbol{\lambda}\right) = \nabla_{\mathbf{u}} \mathbf{J}\left(\mathbf{u}, \mathbf{d}\right) + \nabla_{\mathbf{u}}^{\top} \mathbf{g}\left(\mathbf{u}, \mathbf{d}\right) \boldsymbol{\lambda},$$
$$\mathbf{g}\left(\mathbf{u}, \mathbf{d}\right) = \begin{bmatrix} g_1\left(\mathbf{u}, \mathbf{d}\right) & \ldots & g_{n_{\mathbf{g}}}\left(\mathbf{u}, \mathbf{d}\right) \end{bmatrix}^{\top},$$
$$\boldsymbol{\lambda} = \begin{bmatrix} \lambda_1 & \ldots & \lambda_{n_{\mathbf{g}}} \end{bmatrix}^{\top},$$

Eq. (2.9a) is called stationary condition, Eq. (2.9b) is called primal feasibility condition, Eq. (2.9c) is called dual feasibility condition, and Eq. (2.9d) is called complementary slackness condition [43].

Master Controllers

We can see problem (2.8) as a function of the *primal variable* $\mathbf{u}$, and the (sub-) gradient of problem (2.8) w.r.t the dual variables can be expressed as follows [13],

$$\nabla_{\boldsymbol{\lambda}} \boldsymbol{\mathcal{L}}\left(\mathbf{u}, \mathbf{d}, \boldsymbol{\lambda}\right) = \mathbf{g}\left(\mathbf{u}, \mathbf{d}\right) \tag{2.10}$$

This indicates that in order to achieve steady-state optimal condition it is necessary to **control constraint** $g_i(\mathbf{u}, \mathbf{d}) \rightarrow 0$ by manipulating the associated *dual variable* λ_i. These constraints may be controlled using simple single-loop I-controllers, one for each constraint. This is an excellent feature that enables *transparent* constraint control and easier to tune and operate. This is valid when a constraint g_i is active at optimal steady-state.

The aforementioned idea is also known as subgradient updates in many optimization literatures [13, 44]. The convergence of subgradient updates is only guaranteed for convex problems and for sufficiently small update steps, which also trigger the study in Chapter 3.

According to Theorem 2.3 i.e., dual feasibility condition (2.9c), $\lambda_i \geq 0$ must hold for inequality constraints in problem (2.1). This requirement is ensured by an I-controller, complemented with `max` selectors, one for each I-controller. This structure indicates that the presence of a master controller as *a constraint controller* enables automatic active constraint changing [41]. Thus, this method is flexible in the presence of that changing.

When the constraint switches from the unconstrained to constrained case, the *dual variable* needs to increase from zero to a positive value. In order to do that the master controller needs the constraint g_i to be exceeded over some time, t, in order for the integrator to increase λ_i. Therefore, we must assume that constraint g_i as a soft constraint, $\forall i = 1, \ldots n_{\mathbf{g}}$..

Assumption 2.5: Soft constraint

Constraint violations can be accepted during the iteration over the Lagrange multipliers.

Due to the complementary slackness (2.9d), only one single-loop pairing choice exists.

Assumption 2.6: Single-loop pairing

There is only one single-loop pairing choice, i.e., $g_i \Longleftrightarrow \lambda_i, \forall i = 1, \ldots n_{\mathbf{g}}$.

Assumption 2.7: Perfect (Inner Loop) Control and/or Optimization

In the *inner-loop* control and/or optimization to obtain optimal $\mathbf{u}^\star$ based on steady-state unconstrained optimization problem explained in Section 2.2.2, we have perfect control at each sampling time Δt.

Assumption 2.8: Lipschitz Continuous Gradient

The objective function $J(\mathbf{u}, \mathbf{d})$ has a Lipschitz continuous gradient with a positive constant L, i.e., $||\mathbf{C}^{-1}\nabla_{\mathbf{u}}\mathbf{J}(a) - \mathbf{C}^{-1}\nabla_{\mathbf{u}}\mathbf{J}(b)|| \leq L||a - b||$

Lemma 2.1: Successive Boundedness of the Dual Variables

Suppose Assumption 2.7 and 2.8 hold, then the following inequalities hold

$$||\lambda_i\left(t+1\right)-\lambda_i\left(t\right)||^2 \leq L^2||\mathbf{u}\left(t+1\right)-\mathbf{u}\left(t\right)||^2 \tag{2.11}$$

Proof : Successive Bounded of the Dual Variables

Suppose Assumption 2.7 hold, the the stationary condition at time $t+1$ is given by

$$\nabla_{\mathbf{u}}\mathbf{J}\left(\mathbf{u}\left(t+1\right)\right)+\nabla_{\mathbf{u}}^{\top}g_i\left(\mathbf{u}\left(t+1\right)\right)\lambda_i\left(t+1\right)=\mathbf{0}$$

Assuming linearized constraint g_i,

$$\nabla_{\mathbf{u}}^{\top}g_i\left(\mathbf{u}\left(t+1\right)\right)=-\mathbf{C}$$

$$\lambda_i\left(t+1\right)=\mathbf{C}^{-1}\nabla_{\mathbf{u}}\mathbf{J}\left(\mathbf{u}\left(t+1\right)\right)$$

From Assumption 2.8, we have

$$||\lambda_i\left(t+1\right)-\lambda_i\left(t\right)||=||\mathbf{C}^{-1}\nabla_{\mathbf{u}}\mathbf{J}\left(\mathbf{u}\left(t+1\right)\right)-\mathbf{C}^{-1}\nabla_{\mathbf{u}}\mathbf{J}\left(\mathbf{u}\left(t\right)\right)||$$

$$||\lambda_i\left(t+1\right)-\lambda_i\left(t\right)||\leq L||\left(\mathbf{u}\left(t+1\right)\right)-\left(\mathbf{u}\left(t\right)\right)||$$

from which (2.11) follows. □

Theorem 2.4: Flexibility in Active Constraint Changing

Suppose Assumption 2.1, 2.5, 2.6, 2.7, and 2.8. If there is any change in parameter or disturbance, $\mathbf{d}$, then by utilizing `max` selector, it is possible to decrease and increase the dual variables λ_i to a new non-negative steady-state value.

Proof : Flexibility in Active Constraint Changing

There are four types of active constraint changing, that are (I) unconstrained to unconstrained case, (II) unconstrained to constrained, (III) constrained to unconstrained, and (IV) constrained to constrained.

(I) Unconstrained to unconstrained case: This switch does not need to update the value of λ_i because λ_i is always zero (0), and the presence of `max` selector does not allow negative λ_i.

(II) Unconstrained to constrained case: This switch needs to update the value of λ_i from zero (0) to any positive value. In order to do that, the master controller needs the constraint g_i to be exceeded over some time in order for the integrator to increase the value of λ_i. Using Lemma 2.1, this implies $\lim_{t\to\infty}||\lambda_i\left(t+1\right)-\lambda_i\left(t\right)||=0$, which means that the dual variable λ_i will reach any positive steady-state value.

(III) Constrained to unconstrained case: This switch needs to update the value of λ_i from any positive value to zero (0), and the presence of `max` selector does

not allow negative λ_i.

(IV) Constrained to constrained case, that has two types of case, that are (a) Loose to tight constraint case, and (b) Tight to loose constraint case.

(a) Loose to tight constraint case: This switch needs to update the value of λ_i to larger positive value. In order to do that, the master controller needs the constraint g_i to be exceeded over some time in order for the integrator to increase the value of λ_i.

(b) Tight to loose constraint case: This switch needs to update the value of λ_i to smaller positive value.

Using Lemma 2.1, this implies $\lim_{t\to\infty} ||\lambda_i(t+1) - \lambda_i(t)|| = 0$, which means that the dual variable λ_i of either case (IV.a) or (IV.b) will reach new positive steady-state value. □

Steady-state (Unconstrained-) Optimization Problem

Section 2.2.2 has shown how the *dual variables* are updated and obtained. Therefore, it is possible to see problem (2.8) as a function of *dual variables* $\boldsymbol{\lambda}$, and the *dual variables* as parameter in this modified problem formulation. This section shows 2 possible ways to solve this problem, that are using numerical or equation solver. This is a general strategy.

Steady-state Wolfe Dual Optimization Problem Formulation

Let $\boldsymbol{\lambda} \geq \mathbf{0}$ be the *dual variables* given by the master controllers, then we can see problem (2.8) as unconstrained steady-state optimization problem with Lagrange function as the objective, and the Primal variables (Inputs) as the decision variables. However, the infimum of primal variable $\mathbf{u}$ occurs where the gradient of Lagrange function is equal to zero. Thus, the problem is as follows,

$$\min_{\mathbf{u}} \quad \boldsymbol{\mathcal{L}}(\mathbf{u}, \mathbf{d}, \boldsymbol{\lambda}) \tag{2.12a}$$

$$\text{s.t.} \quad \nabla_{\mathbf{u}} \boldsymbol{\mathcal{L}}(\mathbf{u}, \mathbf{d}, \boldsymbol{\lambda}) = \mathbf{0} \tag{2.12b}$$

where the measurement set is sufficient to estimate the disturbance $\mathbf{d}$ (Assuming sufficient measurement set), and the steady-state gradient estimation is possible if Assumption 2.2 hold. This problem is known as *Wolfe Dual Problem*, that is convenient for computations [45].

To be able to solve problem (2.8), the decision variable, that is primal variable $\mathbf{u}$, should exist in problem (2.12). Therefore, the following assumptions must hold.

Assumption 2.9: The decision variables explicitly appear

Cost and/or constraint function of problem (2.1) is sufficiently continuous and the decision variables explicitly appear in the first derivative of the function(s). For example, at least, twice differentiable such that the decision variables explicitly appear in the gradient of Lagrange function $\nabla_{\mathbf{u}} \boldsymbol{\mathcal{L}}(\mathbf{u}, \mathbf{d}, \boldsymbol{\lambda})$.

If the decision variables do not appear, then constraint (2.12b) is not satisfied, and the solution is infinity (not real number). For example, $\boldsymbol{\mathcal{L}}(\mathbf{u}, \mathbf{d}, \boldsymbol{\lambda}) = 2\mathbf{u}$.

Assumption 2.10: Unique optimal decision variables

Problem (2.12) is guaranteed to have a unique solution within its searching space.

Given $\boldsymbol{\lambda}$, estimated differential state, $\hat{\mathbf{x}}$, and estimated disturbance, $\hat{\mathbf{d}}$, we solve unconstrained problem (2.12) in order to obtain the calculated optimal input $\mathbf{u}^\star$. Any type of numerical solver is acceptable to solve problem (2.12). An input filter is necessary on the calculated optimal *primal variables* $\mathbf{u}^\star$ before implementing the *primal variables* $\mathbf{u}$ in the plant. This filter may avoid too aggressive input implementation. If the solver can provide optimal setpoint of certain output (y^{sp}), it is possible to replace the input filter with a setpoint controller.

Steady-state Equation Problem Formulation

According to Theorem 2.3, the steady-state optimality condition is achieved when *stationary conditions* (2.9a) is satisfied. For fixed values of dual variables, to satisfy the *stationary conditions* (2.9a), we can simply solve the equation set where the gradient of Lagrange function is equal to zero, $\nabla_{\mathbf{u}}\mathcal{L}\left(\mathbf{u},\mathbf{d},\boldsymbol{\lambda}\right) = \mathbf{0}$, for a simple case. For a more complex case, we can formulate an equation problem formulation with the objective function as a norm of the vector $\nabla_{\mathbf{u}}\hat{\mathcal{L}}\left(\mathbf{u},\hat{\mathbf{d}},\boldsymbol{\lambda}\right)$. This results in the following (unconstrained-) optimization problem,

$$\min_{\mathbf{u}} \quad ||\boldsymbol{\nabla}_{\mathbf{u}}\boldsymbol{\mathcal{L}}\left(\mathbf{u},\mathbf{d},\boldsymbol{\lambda}\right)|| \tag{2.13a}$$

$$\text{s.t.} \quad \nabla_{\mathbf{u}}\mathcal{L}\left(\mathbf{u},\mathbf{d},\boldsymbol{\lambda}\right) = \mathbf{0} \tag{2.13b}$$

where the steady-state gradient and paramter/disturbance estimation is possible if assuming sufficient measurement set, and 2.2 hold. Similar to Wolfe Dual, the decision variables explicitly appear if Assumption 2.9 holds.

Given $\boldsymbol{\lambda}$, estimated differential state, $\hat{\mathbf{x}}$, and estimated disturbance, $\hat{\mathbf{d}}$, we solve unconstrained problem (2.13) in order to obtain the calculated optimal input $\mathbf{u}^\star$. Any type of numerical solver is acceptable to solve problem (2.13). One may incorporate input filter on the calculated optimal *primal variables* $\mathbf{u}^\star$ before implementing the *primal variables* $\mathbf{u}$ in the plant.

Remark 2.5: Issues of Dual HRTO

The steady-state optimizer operates on a slow time scale (minutes to hours), while rapid disturbances (milli- to seconds) can render its solution obsolete. Furthermore, Using a numerical solver may pose challenges in numerical robustness, and ensuring the decision variable's presence in the first derivation can limit applicability.

2.3 Feedback-optimizing Control

The optimization layer, referred to as the RTO-layer, operates at a slow time scale (see Figure 2.5-2.6). Due to the potential impact of disturbances at a faster time scale, incorporating some optimization into the control layer is desirable. This ensures that the control layer can guide inputs in the economically correct direction during disturbances, aligning with the concept of feedback-optimizing control [28]. This approach aims to

translate optimization objectives into control objectives. For a detailed exploration of RTO as a feedback control problem, [42] provides a comprehensive review.

Assumptions 2.3 and 2.9 highlight a limitation in utilizing steady-state optimizers for both reduced gradient HRTO and dual HRTO, restricting their applicability. This is evident in scenarios with linear cost functions, necessitating additional techniques like quadratic approximation. In the subsequent sections, the utilization of steady-state optimizers in both approaches is translated into control objectives, transforming them into feedback-optimizing control schemes. Through feedback, the need for Assumptions 2.3 and 2.9 is eliminated, as an initial decision variable guess is provided and continuously updated within the closed-loop system.

2.3.1 Reduced Gradient Feedback-optimizing Control

As previously mentioned, the reduced gradient feedback-optimizing control (FoC) structure has been discussed by [23]. Instead of solving problem (2.6) using steady-state optimizer, this approach utilizes feedback to solve eq. (2.5) as follows,

$$\mathbf{c} = \mathbf{N}^\top \nabla_{\mathbf{u}} \mathbf{J} \left(\mathbf{u}, \mathbf{x}, \mathbf{d} \right) \tag{2.14}$$

In this context, $\mathbf{c} \in \mathbb{R}^{n_\mathbf{u}}$ represents the self-optimizing controlled variables that are targeted to achieve a set-point of zero. This approach eliminates the need for a numerical solver and input filter, as the feedback controller itself can serve as a filter.

This *pure* feedback-optimizing control structure corresponds to the closed-loop implementation with setpoint adjustments, as depicted in Figure 2.2(d). A more comprehensive block diagram of this approach is shown in Figure 2.7, where u can serve as the setpoint for the lower layer controller, such as a setpoint controller. However, it is essential to note that the approach comes with certain challenges, highlighted in Remark 2.6.

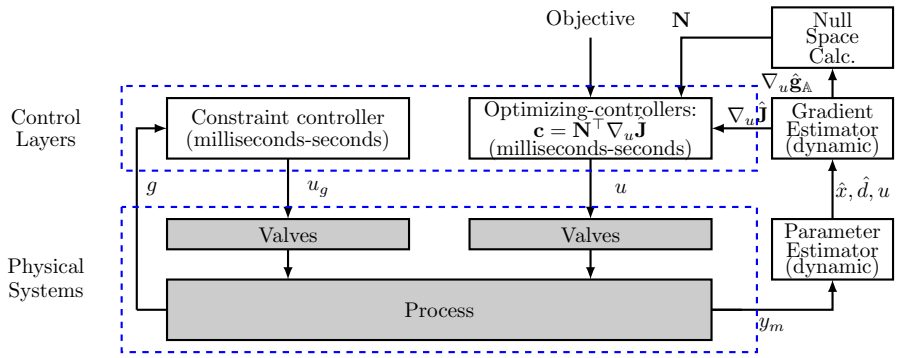


Figure 2.7: Reduced Gradient FOC

Gradient estimators: Both reduced gradient HRTO and FOC need a gradient estimator for active constraint steady-state gradients (see Figure 2.5 and Figure 2.7). Reduced gradient FOC requires a gradient estimator for cost function steady-state gradients (see Figure 2.7), while reduced gradient HRTO does not (see Figure 2.5).

Remark 2.6: Issues of Reduced Gradient FOC

Similar to reduced gradient HRTO, this approach requires the identification of all possible nullspaces corresponding to the various combinations of active constraints. For processes with a large number of constraints, this task can become impractical.

2.3.2 Primal Dual Feedback-optimizing Control

As previously discussed by [16], the *general* primal-dual feedback-optimizing control (FoC) structure can be applied to interactive systems. Instead of using a steady-state optimizer to solve problem (2.12) or (2.13), it is possible to solve the equation set (2.9a) using feedback control to translate the entire optimization problem into *pure* feedback control problem:

$$\mathbf{c} = \nabla_{\mathbf{u}} \mathcal{L}\left(\mathbf{u}, \mathbf{d}, \boldsymbol{\lambda}\right) \tag{2.15}$$

Here, $\mathbf{c} \in \mathbb{R}^{n_{\mathbf{u}}}$ represents the self-optimizing controlled variables targeted to achieve a set-point of zero. The controller is also called gradient controller.

Aligned with [12], controlling the Lagrangian function's gradient ($\nabla_{\mathbf{u}} \mathcal{L}(\mathbf{u}, \mathbf{d}, \boldsymbol{\lambda})$) to zero is essential to satisfy the stationary condition of the necessary condition of optimality (NCO). Assuming an optimal solution exists, we treat the Lagrangian function's gradient as self-optimizing controlled variables, and PID controllers are employed to drive $\nabla_{\mathbf{u}} \mathcal{L}(\mathbf{u}, \mathbf{d}, \boldsymbol{\lambda})$ to zero.

This approach, like reduced gradient FoC, offers benefits by eliminating numerical solvers and input filters. It aligns with the closed-loop implementation with setpoint adjustments, depicted in Figure 2.2(d). The corresponding block diagram in Figure 2.8 illustrates the approach, with u can serve as the setpoint for the lower layer controller.

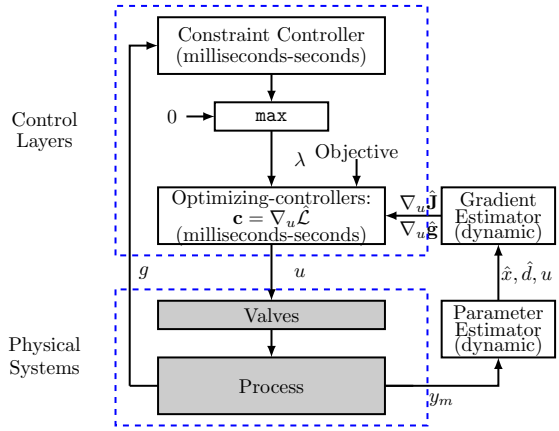


Figure 2.8: Primal Dual FoC

Primal-dual FoC, like dual-based HRTO, excels in automatic handling of active constraint switching—an ability absent in reduced gradient FoC. Yet, it introduces a distinct challenge, elaborated in Remark 2.7. Chapter 3 addresses and resolves this challenge.

Gradient estimators: Primal dual FoC necessitates a gradient estimator for steady-state gradients of both cost and constraint functions (see Figure 2.8), whereas dual-based HRTO does not (see Figure 2.6).

Number of required controllers: The number of required controllers in primal-dual FoC approach is $n_{\mathbf{u}} + n_{\mathbf{g}}$.

Remark 2.7: Issues of Primal Dual FoC

Controlling constraints in Primal-Dual FoC can be challenging because the constraint being controlled at a relatively slower time scale than the optimizing/gradient controllers, posing difficulties in tight constraint control.

2.3.3 Regarding Interactive Systems

Both reduced gradient FoC and primal-dual FoC require proper pairing procedure, as outlined in [46] where the Relative Gain Array (RGA) plays a crucial role. However, in interactive systems, pairing is challenging, leading to negative RGA elements and potential instability. Primal-dual FoC has more consistent interaction structure, and is preferred for handling active constraint switching due to its *flexibility*. Thus, this section will focus on explaining the interactive system based on the primal-dual FoC structure.

First, consider a linearized gain matrix $\mathbf{G}$ form $\mathbf{u}$ to $\nabla_{\mathbf{u}}\mathcal{L}$

$$\nabla_{\mathbf{u}}\mathcal{L} = \mathbf{G}\mathbf{u} \tag{2.16}$$

To be more precise,

$$\begin{bmatrix} \nabla_{\mathbf{u}}\mathcal{L}(1) \\ \vdots \\ \vdots \\ \nabla_{\mathbf{u}}\mathcal{L}(n_{\mathbf{u}}) \end{bmatrix} = \begin{bmatrix} G_{1,1} & G_{1,2} & \dots & G_{1,n_{\mathbf{u}}} \\ G_{2,1} & G_{2,2} & & G_{2,n_{\mathbf{u}}} \\ \vdots & & \ddots & \vdots \\ G_{n_{\mathbf{u}},1} & G_{n_{\mathbf{u}},2} & \dots & G_{n_{\mathbf{u}},n_{\mathbf{u}}} \end{bmatrix} \begin{bmatrix} u_1 \\ \vdots \\ \vdots \\ u_{n_{\mathbf{u}}} \end{bmatrix}$$

Based on equation (2.16), three distinct systems can be defined as follows.

Definition 2.4: Interactive Systems

- Decomposed (decoupled) system: In this system, $G_{i,j} = 0$ for $i \neq j$, meaning that the non-diagonal elements of matrix $\mathbf{G}$ are zero. As a result, there is no influence of input u_i on $\nabla_{\mathbf{u}}\mathcal{L}(j)$, and vice versa.
- Weakly interactive systems: This system exhibits $G_{i,j} \approx 0$ for $i \neq j$, indicating that the non-diagonal elements of $\mathbf{G}$ are close to zero. Consequently, there is only insignificant influence of input u_i on $\nabla_{\mathbf{u}}\mathcal{L}(j)$ or vice versa.
- Highly interactive systems: In this system, $G_{i,j} \neq 0$ for $i \neq j$, signifying that the non-diagonal elements of matrix $\mathbf{G}$ are non-zero. As a result, the influence of input u_i may be significant enough to destabilize $\nabla_{\mathbf{u}}\mathcal{L}(j)$ or vice versa.

In a decomposed (decoupled) system or weakly interactive systems, single-loop controllers (e.g., I-controllers) can be employed to drive $\nabla_{\mathbf{u}}\mathcal{L}(i)$ to zero, utilizing the pairing $\nabla_{\mathbf{u}}\mathcal{L}(i) \longleftrightarrow u_i$, which is obvious, and single-loop controllers work well [40, 41, 47, 48].

In highly interactive systems, selecting the pairing for gradient controllers becomes less straightforward. One approach is using the RGA method, but this might necessitate a good model and, in some cases, a suitable pairing might not be possible. Note that highly interactive systems do not always imply instability. The instability might occurs because unsuitable gradient control pairing. Another option is the dual-based HRTO, as described in Section 2.2.2.

Remark 2.8: Time-varying parameter

Time-varying parameters can lead to interaction in two ways: indirect, when a local time-varying parameter influences other subsystems through its local solution changes (i.e., $\nabla_{u_i}\mathcal{L} = f_i(u_i, u_j(p_j), \lambda)$), and direct, when a local time-varying parameter directly affects other subsystems' solutions (i.e., $\nabla_{u_i}\mathcal{L} = f_i(u_i, p_j, \lambda)$).

2.4 Numerical Examples

2.4.1 Isothermal CSTR with quadratic cost function

In this section, we examine the optimal operation of an isothermal Continuously Stirred Tank Reactor (CSTR) involving two exothermic reactions [49, 50]:

$$A + B \rightarrow C \tag{2.17a}$$
$$2B \rightarrow D \tag{2.17b}$$

The objective is to produce the desired product, C, while minimizing the undesired by-product, D. The CSTR is supplied by two feed streams, $u_1 = F_A$ and $u_2 = F_B$, with known inlet concentrations C_{A_i} and C_{B_i} respectively.

In the case of perfect temperature control (isothermal) and constant level (V) control, the steady-state model of the process is described by the following component mass balances:

$$\mathbf{f}_1 : F_A C_{A_i} - (F_A + F_B)C_A - k_1 C_A C_B V = 0 \tag{2.18a}$$
$$\mathbf{f}_2 : F_B C_{B_i} - (F_A + F_B)C_B - k_1 C_A C_B V - 2k_2 C_B^2 V = 0 \tag{2.18b}$$
$$\mathbf{f}_3 : -(F_A + F_B)C_C + k_1 C_A C_B V = 0 \tag{2.18c}$$

where C_A, C_B, and C_C are the concentration of component A, B, and C, respectively. Further, k_1 and k_2 are the kinetic coefficient of reaction 1 and 2, respectively. Finally, V is the tank volume.

The heat produced by the chemical reaction is given by,

$$\mathbf{f}_4 : Q = (-\Delta H_1)k_1 C_A C_B V + (-\Delta H_2)k_2 C_B^2 V \tag{2.19}$$

where ΔH_1 and ΔH_2 denotes the enthalpy of the reactions 1 and 2, respectively. This section exclusively considers the steady-state model, excluding system dynamics, *in order to specifically analyze controller dynamics among different approaches.* The nominal model parameters are the same as the one used in [50], and are in Table. 2.1.

The goal is to maximize the amount of product C, represented by the expression $(F_A + F_B)\,C_C$ multiplied by the yield factor $\frac{(F_A+F_B)C_C}{F_A C_{A_i}}$. Maximum bounds are set for

the heat produced by the reactions, denoted as Q, and the total flow, defined as $F := F_A + F_B$. The optimization problem can be formulated as follows:

$$\min_{F_A, F_B} \quad J = -\frac{(F_A + F_B)^2 C_C^2}{F_A C_{A_i}} \tag{2.20a}$$

$$\text{s.t.} \quad \mathbf{f}_1 - \mathbf{f}_4 : \text{Equations (2.18)-(2.19)}, \tag{2.20b}$$

$$\mathbf{g}_1 : \frac{Q}{Q_{max}} - 1 \leq 0, \tag{2.20c}$$

$$\mathbf{g}_2 : \frac{F_A + F_B}{F_{max}} - 1 \leq 0 \tag{2.20d}$$

The reaction rate k_1 varies from 0.3 to 1.5 $\frac{l}{mol\ hr}$, leading to changes in the optimal feed rates, objective function, and active constraints in the optimal solution.

Table 2.1: Nominal model parameters and parameter bounds

Parameters	Value	Unit
C_{A_i}	2	$\frac{mol}{l}$
C_{B_i}	1.5	$\frac{mol}{l}$
k_1	1.5	$\frac{l}{mol\ hr}$
k_2	0.014	$\frac{l}{mol\ hr}$
ΔH_1	-7×10^4	$\frac{J}{mol}$
ΔH_2	-1×10^5	$\frac{J}{mol}$
V	500	l
F_{max}	22	$\frac{l}{hr}$
Q_{max}	1×10^6	$\frac{J}{hr}$

Since we have $n_{\mathbf{g}} = 2$ constraints, there are, at maximum, four potential active constraint regions ($2^2 = 4$):

1. Fully unconstrained (unlikely)
2. Only g_1 active (R-I)
3. Only g_2 active (R-II)
4. Both g_1 and g_2 active (R-III)

To verify potential active constraints, we perform offline numerical optimization for expected disturbances, confirming that only three different combinations of active constraints need consideration. Figure 2.9 depicts the active constraint regions, labeled R-I, R-II, and R-III, as a function of k_1.

Reduced gradient approaches

As depicted in Figure 2.9, there are three sets of active constraints. Consequently, when implementing reduced gradient approaches, it is necessary to create three distinct control structures, each corresponding to one of the active constraint regions.

Region I (R-I): In this region, the maximum heat temperature $(\mathbf{g}_1)$ requires tight control. Following the pairing rule proposed by [46], we control $\mathbf{g}_1$ by manipulating F_A $(\mathbf{g}_1 \leftrightarrow F_A)$. See PID-1 in Figure 2.10.

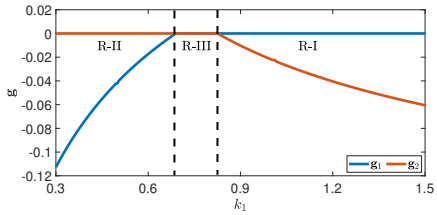


Figure 2.9: Active constraints at the optimum changes with k_1

Utilizing Theorem 2.2, we numerically solve the unconstrained steady-state optimization problem by determining the optimal remaining degree of freedom, F_B. The problem is formulated as follows:

$$\min_{F_B} \quad ||\mathbf{N}_1^\top \nabla_{\mathbf{u}} \mathbf{J}\,(\mathbf{u}, \mathbf{d})\,|| \tag{2.21}$$

where $\mathbf{N}_1$ represents the nullspace of $\nabla_{\mathbf{u}} \mathbf{g}_1\,(\mathbf{u}, \mathbf{d})$.

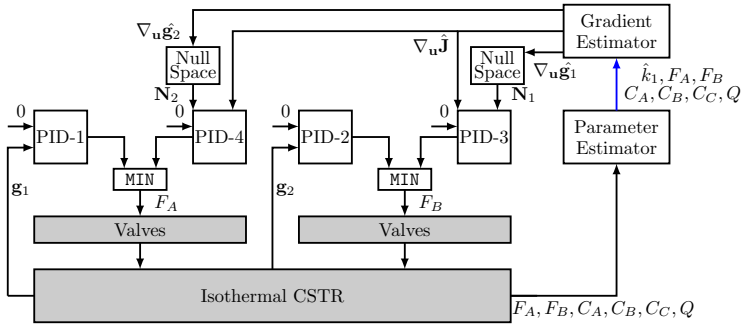


Figure 2.10: Reduced gradient FOC applied to Isothermal CSTR

Region II (R-II): In this region, tight control of the maximum total flowrate ($\mathbf{g}_2$) is essential. Manipulating F_B ($\mathbf{g}_2 \leftrightarrow F_B$) tightly regulates this constraint. See PID-2 in Figure 2.10. The remaining degree of freedom (F_A) is utilized to solve the following unconstrained steady-state optimization problem:

$$\min_{F_A} \quad ||\mathbf{N}_2^\top \nabla_{\mathbf{u}} \mathbf{J}\,(\mathbf{u}, \mathbf{d})\,|| \tag{2.22}$$

Here, $\mathbf{N}_2$ represents the nullspace of $\nabla_{\mathbf{u}} \mathbf{g}_2(\mathbf{u}, \mathbf{d})$.

Region III (R-III): In this region, tight control is required for both the maximum heat temperature ($\mathbf{g}_1$) and the maximum total flow rate ($\mathbf{g}_2$). As a result, both degrees of freedom are utilized to ensure tight control over both constraints. Following the pairing rule proposed by [46], we control $\mathbf{g}_1$ by manipulating F_A ($\mathbf{g}_1 \leftrightarrow F_A$) and control $\mathbf{g}_2$ by manipulating F_B ($\mathbf{g}_2 \leftrightarrow F_B$).

Solving numerical optimization problem using feedback: As discussed in Section 2.3.1, both problem (2.21) and (2.22) can be solved using feedback, also known as reduced gradi-

ent FOC, with the following pairings:

$$\mathbf{c}_1 : \mathbf{N}_1^\top \nabla_{\mathbf{u}} \mathbf{J}\left(\mathbf{u}, \mathbf{d}\right) \leftrightarrow F_B \text{ (See PID-3 in Figure 2.10)} \tag{2.23}$$

$$\mathbf{c}_2 : \mathbf{N}_2^\top \nabla_{\mathbf{u}} \mathbf{J}\left(\mathbf{u}, \mathbf{d}\right) \leftrightarrow F_A \text{ (See PID-4 in Figure 2.10)} \tag{2.24}$$

Region Switching: To switch between the different active constraint regions, we employ `min` selector blocks [51], as depicted in Figure 2.10.

Controllers parameters: The PID controllers are tuned using the SIMC tuning rule [52]. The controllers parameters used in the reduced gradient FOC are listed in Table 2.2.

Table 2.2: Parameters used in the controllers and filters shown in Figure 2.10

Tuning par.	PID-1	PID-2	PID-3	PID-4
K_I	4.3437	7.3333	2.9936	-0.8014

Primal dual approaches

It is *unnecessary* to identify active constraint regions in implementing primal dual approaches. We only need to design constraint controllers as master controllers, and solve the unconstrained steady-state optimization problem described as follows.

Master controllers: There are two constraint controllers, equipped with `max` selector in each, with the following pairings:

$$\mathbf{g}_1 \leftrightarrow \tilde{\lambda}_1 \text{ (See PID-1 in Figure 2.11)}; \qquad \lambda_1 = \texttt{max}\left(0, \tilde{\lambda}_1\right); \tag{2.25}$$

$$\mathbf{g}_2 \leftrightarrow \tilde{\lambda}_2 \text{ (See PID-2 in Figure 2.11)}; \qquad \lambda_2 = \texttt{max}\left(0, \tilde{\lambda}_2\right); \tag{2.26}$$

To ensure 'good' control, these controllers are 5 times slower than the gradient controllers.

Steady-state unconstrained optimization problem: Utilizing Theorem 2.3, we numerically solve the unconstrained steady-state optimization problem by determining the optimal decision variables (primal variables), F_A and F_B. The problem is formulated as follows:

$$\min_{F_A, F_B} \quad ||\nabla_{\mathbf{u}} \mathcal{L}\left(\mathbf{u}, \mathbf{d}, \boldsymbol{\lambda}\right)|| \tag{2.27}$$

where $\mathcal{L}\left(\mathbf{u}, \mathbf{d}, \boldsymbol{\lambda}\right) = J\left(\mathbf{u}, \mathbf{d}\right) + \lambda_1 \mathbf{g}_1 + \lambda_2 \mathbf{g}_2$, and $\nabla_{\mathbf{u}} \mathcal{L}\left(\mathbf{u}, \mathbf{d}, \boldsymbol{\lambda}\right) = \begin{bmatrix} \nabla_{\mathbf{u}} \mathcal{L}(1) & \nabla_{\mathbf{u}} \mathcal{L}(2) \end{bmatrix}^\top$.

Solving numerical optimization problem using feedback: In Section 2.3.2, problem (2.27) is solvable using feedback (primal dual FOC). The linearized gain matrix $\mathbf{G}$ for this problem is:

$$\mathbf{G} = \begin{bmatrix} 0.2984 & -0.1455 \\ -0.1412 & 0.0722 \end{bmatrix}$$

which indicates that the system is interactive. Though the pairing is not obvious, fortunately, RGA analysis recommends the following diagonal pairings:

$$\mathbf{c}_1 : \nabla_{\mathbf{u}} \mathcal{L}(1) \leftrightarrow F_A \text{ (See PID-3 in Figure 2.11)} \tag{2.28}$$

$$\mathbf{c}_2 : \nabla_{\mathbf{u}} \mathcal{L}(2) \leftrightarrow F_B \text{ (See PID-4 in Figure 2.11)} \tag{2.29}$$

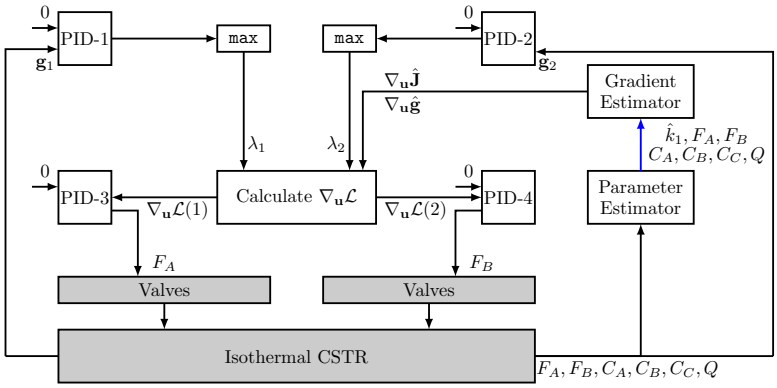


Figure 2.11: Primal dual FOC applied to Isothermal CSTR

For cases where *off-diagonal* pairings are recommended, refer to [16] and see Figure 2.12 as illustration. This justify the use of equation solver for a highly interactive system.

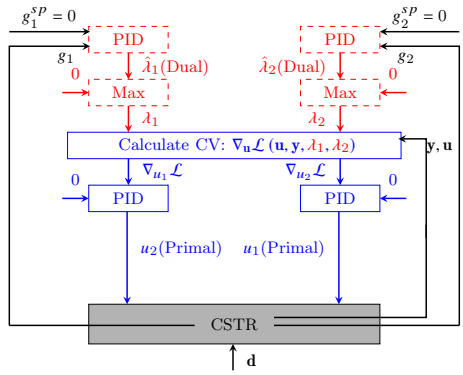


Figure 2.12: Primal dual FOC applied to Isothermal CSTR with *off-diagonal* case.

Controllers parameters: The PID controllers are tuned with parameters from Table 2.3.

Table 2.3: Parameters used in the controllers and filters shown in Figure 2.11

Tuning par.	PID-1	PID-2	PID-3	PID-4
K_P	-5.4797	-4.7433	–	–
K_I	-0.0415	-0.0359	1.1172	4.6172

Simulator and estimators

The Isothermal CSTR simulator is developed using the `CasADi ver. 3.5.1` toolbox [53] in `MATLAB R2019b`, and simulated using the `rootfinder` function. The resulting NLP problems are solved using `IPOPT v3.12.2`. Simulations are performed on a `2.11 GHz` processor with `16 GB` memory. Parameter $\hat{k}_1$ is estimated using a model-based approach (see Equation 2.19), with the gradient estimated analytically through the model-based approach using `CasADi`, a self-contained symbolic framework.

Simulation results

In this simulation, we compare the performance of reduced gradient FOC, primal dual FOC, and a standard HRTO as a benchmark (Figure 2.13). The results show that approaches with transparent constraint control can reach the optimal steady-state solution effectively, making them powerful alternatives for HRTO. However, as expected, primal dual FOC does not control the constraints as tight as reduced gradient FOC.

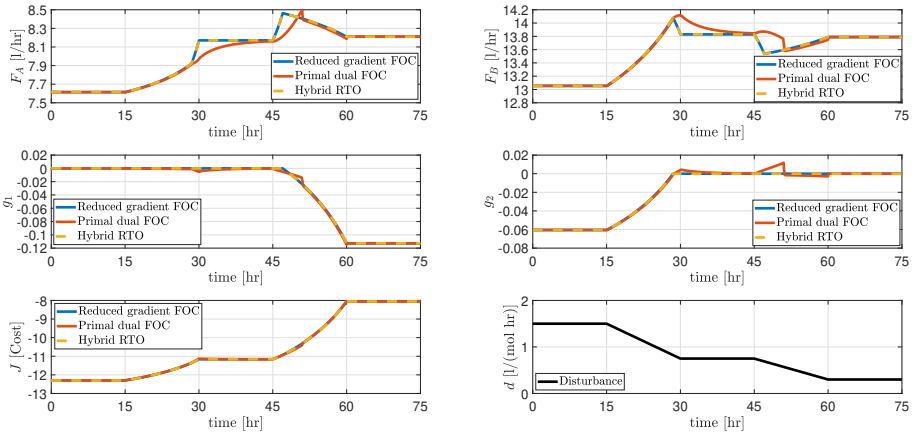


Figure 2.13: Simulation results comparing reduced gradient, primal-dual FOC, and standard HRTO.

Notice that both problem (2.21) and (2.22) satisfy Assumption 2.3 because F_B, and F_A, respectively, still exist after the first derivation of the objective function, as shown below:

$$\begin{bmatrix} \frac{\partial J}{\partial F_A} \\ \frac{\partial J}{\partial F_B} \end{bmatrix} = - \begin{bmatrix} \frac{2C_{A_i}C_C F_A(F_A+F_B)\left(C_C+(F_A+F_B)\frac{\partial C_C}{\partial F_A}\right)-C_{A_i}(F_A+F_B)^2C_C^2}{\left(C_{A_i}F_A\right)^2} \\ \frac{2C_{A_i}C_C F_A(F_A+F_B)\left(C_C+(F_A+F_B)\frac{\partial C_C}{\partial F_A}\right)}{\left(C_{A_i}F_A\right)^2} \end{bmatrix} \tag{2.30}$$

Therefore, the implementation of reduced gradient HRTO is possible as shown in Figure 2.14. The direction of the input filter depends on the gain direction between the selected manipulated variable (MV) and the reduced gradient (CV). Here, the gain directions from F_A to $\mathbf{c}_2$ and F_B to $\mathbf{c}_1$ are opposite, resulting in a negative direction for the input filters labeled as F3 and F4 in Figure 2.14. A complete systematic approach to tune the

input filters are not so clear as far as the author knowledge. The controllers parameters used in the reduced gradient HRTO are listed in Table 2.4.

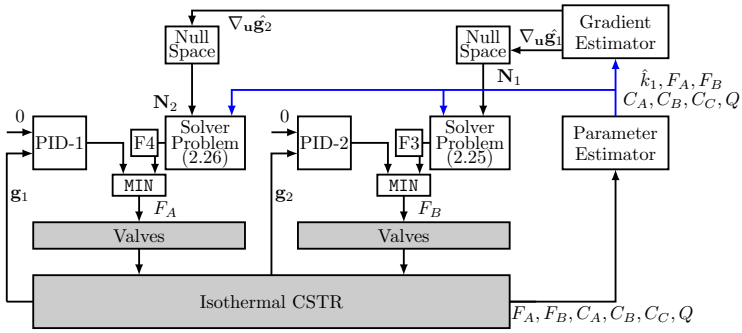


Figure 2.14: Reduced gradient HRTO applied to Isothermal CSTR

Table 2.4: Parameters used in the controllers and filters shown in Figure 2.14

Tuning par.	PID-1	PID-2	F3	F4
K_I	4.3437	7.3333	–	–
K_u	–	–	−0.0385	−0.0385

While process control prioritizes factors beyond solver computation time, it's essential to highlight that reduced gradient HRTO exhibits faster computation times than standard HRTO (approximately 38% reduction), as indicated in Table 2.5. This is due to the ease of solving an (unconstrained) optimization problem in contrast to a constrained one. Notably, reduced gradient FOC is excluded from the comparison, as steady-state solvers are inherently unnecessary for them.

In this simulation, we compare HRTO and reduced gradient HRTO, using two solvers (in R-I and R-II). Despite concerns about employing two solvers, the average computation time surpasses standard HRTO. An alternative is presented by the dual HRTO approach, as exemplified in [16], offering a solution.

Table 2.5: Average Computation Time

Approaches	Comp. Time [Sec]	Iterations
Reduced gradient HRTO solver (ave.)	0.0250	6.4252
Reduced gradient HRTO solver (2.21)	0.0217	5.6046
Reduced gradient HRTO solver (2.22)	0.0282	7.2457
HRTO solver	0.0406	9.0697

2.4.2 Interactive gas-lift optimization with linear cost function

This section showcases the use of reduced gradient and primal-dual FOC for optimizing an interactive gas-lifted oil production network with two wells. As illustrated in Figure 2.15, these wells jointly produce hydrocarbons to a common riser manifold (see

Appendix A). The optimization problem is stated as follows:

$$\min_{w_{l,1},\, w_{l,2}} \quad J = -p_o w_o + p_l \sum_{i=1}^{2} w_{l,i} \tag{2.31a}$$

$$\text{s.t.} \quad \mathbf{f}(\mathbf{x}, \mathbf{z}, \mathbf{u}, \mathbf{d}) = \mathbf{0}, \tag{2.31b}$$

$$\mathbf{h}(\mathbf{x}, \mathbf{z}, \mathbf{u}, \mathbf{d}) = \mathbf{0}, \tag{2.31c}$$

$$\mathbf{g} : w_g - \bar{w}_g \leq 0 \tag{2.31d}$$

In the above formulation, p_o and p_l denote oil price and gas lift treatment cost, respectively. Moreover, $w_o = \sum_{i=1}^{2} w_{o,i}$, $w_g = \sum_{i=1}^{2} w_{g,i}$, and $w_l = \sum_{i=1}^{2} w_{l,i}$, where $w_{o,i}$ and $w_{g,i}$ are oil and gas exports from well i. Gas-lift rate, $w_{l,i}$, serves as manipulated variables, and $\bar{w}_g$ represents maximum gas processing capacity. Variables $\mathbf{x}$, $\mathbf{z}$, $\mathbf{u}$, and $\mathbf{d}$ represent differential states, algebraic states, inputs, and disturbances, respectively, with $\mathbf{f}(\cdot)$ and $\mathbf{g}(\cdot)$ denoting differential and algebraic functions.

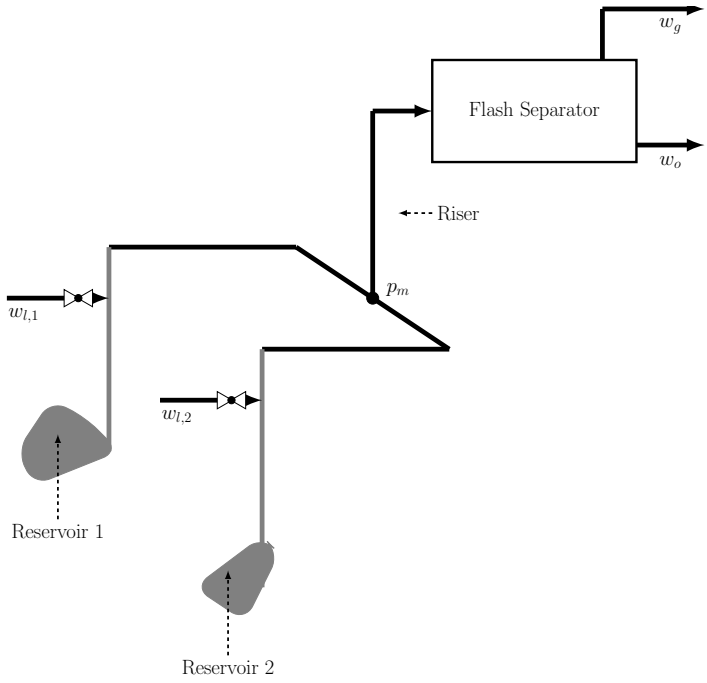


Figure 2.15: Interactive Gas Lift Optimization Schematic Drawing.

Changes in gas-lift rates of one well impact others as hydrocarbons mix in the riser, creating system interactivity. Furthermore, objective function (2.31a) is linear, rendering reduced gradient HRTO and dual-based HRTO inapplicable without further adjustments.

Reduced gradient FOC

With just a single constraint (2.31d), we can identify two distinct control structures, each corresponding to one of the active constraint regions: unconstrained and constrained.

Region I (R-I): In this region, where constraint $\mathbf{g}$ is active, one MV is dedicated to tightly control constraint $\mathbf{g}$, while the other MVs are employed to drive the reduced gradient towards zero. As proposed by [46], we establish the following pairing:

$$\mathbf{g} \leftrightarrow w_{l,2} \text{ (See PID-1 in Figure 2.16)} \tag{2.32}$$

$$\mathbf{c} : \mathbf{N}^{\top}\nabla_{\mathbf{u}}\mathbf{J} \leftrightarrow w_{l,1} \text{ (See PID-2 in Figure 2.16)} \tag{2.33}$$

Region II (R-II): In this region, where constraint $\mathbf{g}$ is inactive, all MVs are utilized to drive the gradient $\left(\nabla_{\mathbf{u}}\mathbf{J} \in \mathbb{R}^2\right)$ to zero. As proposed by [46], the pairing is as follows:

$$\mathbf{c}_1 : \nabla_{\mathbf{u}}\mathbf{J}(1) \leftrightarrow w_{l,1} \text{ (See PID-3 in Figure 2.16)} \tag{2.34}$$

$$\mathbf{c}_2 : \nabla_{\mathbf{u}}\mathbf{J}(2) \leftrightarrow w_{l,2} \text{ (See PID-4 in Figure 2.16)} \tag{2.35}$$

where $\nabla_{\mathbf{u}}\mathbf{J} = \begin{bmatrix}\nabla_{\mathbf{u}}\mathbf{J}(1) & \nabla_{\mathbf{u}}\mathbf{J}(2)\end{bmatrix}^{\top}$.

Figure 2.16 depicts the reduced gradient FOC structures for this problem. The PID

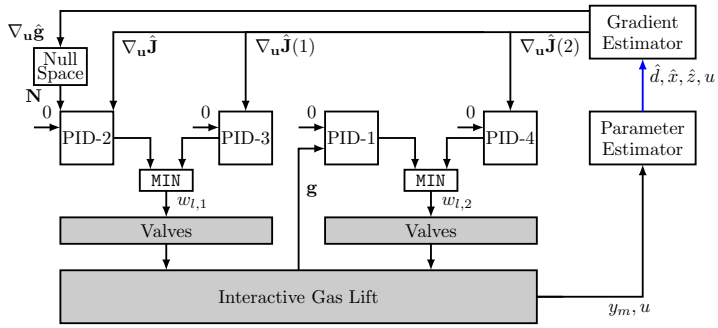


Figure 2.16: Reduced gradient FOC applied to Interactive Gas Lift

controllers are tuned using the SIMC tuning rule [52]. The controllers parameters used in the reduced gradient FOC are listed in Table 2.6.

Table 2.6: Parameters used in the controllers shown in Figure 2.16

Tuning par.	PID-1	PID-2	PID-3	PID-4
K_P	3.6425	−16.0234	12.7008	12.9354
K_I	0.0064	−0.0267	0.0212	0.0216

Primal dual FOC

Figure 2.17 depicts the reduced gradient FOC structures for this problem.

Master controllers: There is a constraint controller, equipped with `max` selector:

$$\mathbf{g} \leftrightarrow \tilde{\lambda} \text{ (See PID-1 in Figure 2.17)}; \qquad \lambda = \texttt{max}\left(0, \tilde{\lambda}\right); \tag{2.36}$$

To ensure 'good' control, this controller is 5 times slower than the gradient controllers.

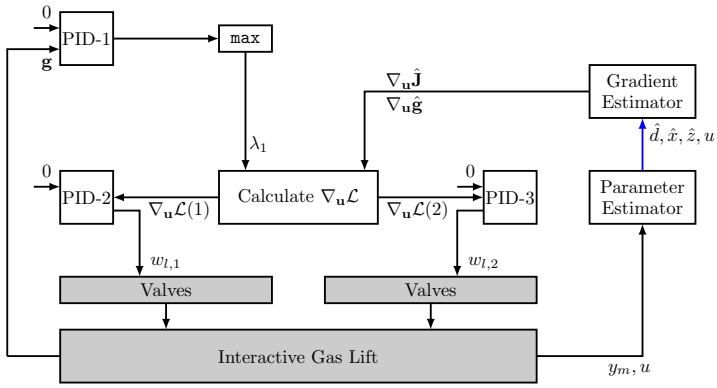


Figure 2.17: Primal dual FOC applied to Interactive Gas Lift

Steady-state unconstrained optimization problem: Utilizing Theorem 2.3, the problem is formulated as follows:

$$\min_{w_{l,1},\, w_{l,2}} \quad ||\nabla_{\mathbf{u}}\mathcal{L}\left(\mathbf{u}, \mathbf{d}, \boldsymbol{\lambda}\right)|| \tag{2.37}$$

where $\mathcal{L}\left(\mathbf{u}, \mathbf{d}, \boldsymbol{\lambda}\right) = J\left(\mathbf{u}, \mathbf{d}\right) + \lambda \mathbf{g}$, and $\nabla_{\mathbf{u}}\mathcal{L}\left(\mathbf{u}, \mathbf{d}, \boldsymbol{\lambda}\right) = \begin{bmatrix}\nabla_{\mathbf{u}}\mathcal{L}(1) & \nabla_{\mathbf{u}}\mathcal{L}(2)\end{bmatrix}^{\top}$.

Solving numerical optimization problem using feedback: In Section 2.3.2, problem (2.37) is solvable using feedback (primal dual FOC). The linearized gain matrix $\mathbf{G}$ is:

$$\mathbf{G} = \begin{bmatrix}0.3654 & 0.0205\\ 0.0205 & 0.3637\end{bmatrix}$$

which indicates that the system is weakly interactive, where the pairing is obvious, and RGA analysis recommends diagonal pairings:

$$\mathbf{c}_1 : \nabla_{\mathbf{u}}\mathcal{L}(1) \leftrightarrow w_{l,1} \text{ (See PID-2 in Figure 2.17)} \tag{2.38}$$

$$\mathbf{c}_2 : \nabla_{\mathbf{u}}\mathcal{L}(2) \leftrightarrow w_{l,2} \text{ (See PID-3 in Figure 2.17)} \tag{2.39}$$

The controllers parameters used in the primal dual FOC are listed in Table 2.7.

Table 2.7: Parameters used in the controllers shown in Figure 2.17

Tuning par.	PID-1	PID-2	PID-3
K_P	-0.0240	11.9688	11.1624
K_I	-0.0001	0.0199	0.0186

Simulator and estimators

The gas lift simulator, developed in `MATLAB R2019b` using the `CasADi ver. 3.5.1` ([53]), employs the `IDAS` integrator. Each well is assumed to have either a Multiphase flow meter (MPFM) or virtual flow metering (VFM) with soft sensors for estimating disturbance $\hat{d}$, differential states $\hat{x}$, and algebraic states $\hat{z}$ as needed [54]. Additionally, gradient estimation is conducted via a model-based approach [55].

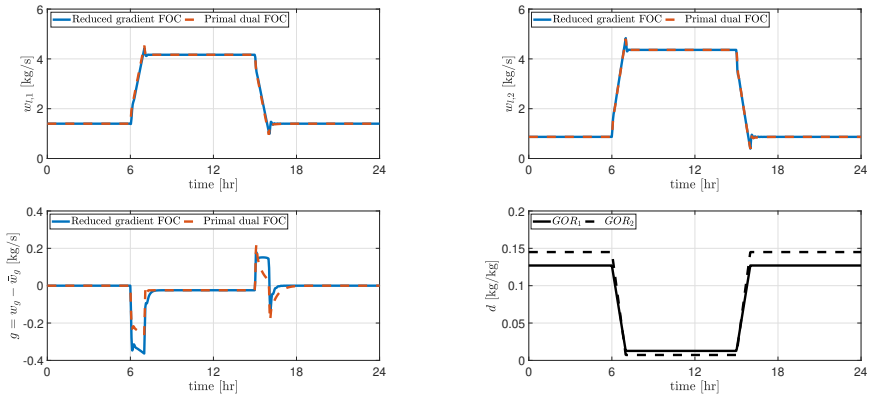


Figure 2.18: Simulation results comparing the implementation of reduced gradient FOC and primal-dual FOC in an interactive gas lifted oil production system.

Simulation results

In this simulation, we compare reduced gradient FOC with primal dual FOC (see Figure 2.18). Results indicate transparent constraint control approaches effectively achieve optimal steady-state solutions, offering potential alternatives for Hybrid RTO. Process dynamics influence constraint violations; although primal dual FOC controls the constraints on a slow time scale, maximum dynamic violations decrease due to the process dynamics. However, their effects aren't consistently positive and may lead to adverse outcomes depending on disturbance trajectories. Notably, primal-dual FOC's maximum dynamic constraint violation slightly exceeds reduced gradient FOC's. Despite its importance, primal-dual FOC *indirectly* controls constraints on a slow time scale, resulting in lower constraint satisfaction compared to reduced gradient FOC.

2.5 Chapter Summary

In this chapter, we discussed several approaches to achieve optimal performance, namely

- S-I: Steady-state RTO with dynamic model adaptation (also known as Hybrid RTO)
- S-II: Traditional Self-optimizing Control
- S-III: Reduced gradient hybrid RTO
- S-IV: Dual-based hybrid RTO
- S-V: Reduced gradient FOC
- S-VI: Primal dual FOC

The parameters we assess and compare are:

- (a): Is steady-state optimal achieved?
- (b): Is constraint controlled *transparently* (see Definition 2.1)?
- (c): Is flexible for active constraint switching?
- (d): Is constraint controlled *directly* (on fast time scale)?
- (e): Is applicable for less than twice differentiable Lagrange function?

- (f): Is numerical solver avoidable (computation time)?
- (g): Is it recommended for complex and large system?
- (h): Is input filter (or additional setpoint controller) avoidable?
- (i): Is parameters and states dynamic estimator avoidable?
- (j): Is gradient estimator avoidable?

The discussion's essence is summarized in Table 2.8, where traditional self-optimizing control emerges as the preferred choice among considered *success* parameters, despite its *significant* limitation of near-optimal performance. This leads us to the conclusion that both reduced gradient FOC and primal dual FOC are highlighted as superior alternatives, *transparently* controlling constraints and avoiding the need for twice differentiable Lagrange functions. However, while reduced gradient FOC lacks adaptability to active constraint switching, our attention now turns to improving the practical utility of primal-dual FOC.

Table 2.8: Summary of Chapter 2

	S-I	S-II	S-III	S-IV	S-V	S-VI
(a)	Yes	No	Yes	Yes	Yes	Yes
(b)	No	Yes	Yes	Yes	Yes	Yes
(c)	Yes	No	No	Yes	No	Yes
(d)	No	Yes	Yes	No[1]	Yes	No[1]
(e)	Yes	Yes	No	No	Yes	Yes
(f)	No (High)	Yes	No (Medium)	No (Medium)	Yes	Yes
(g)	Yes	No	Yes	Yes	No	No[2]
(h)	No	Yes	No	No	Yes	Yes
(i)	No	Yes	No	No	No	No
(j)	Yes	Yes	No	Yes	No	No

"Yes": the approach has satisfied the success parameter
"No": the approach does not satisfied the success parameter
"No": the approach does not satisfied the success parameter, and will be addressed in this thesis.
[1]: will be addressed in Chapter 3
[2]: will be addressed in Chapter 4
"No (High)": the approach does not satisfied the success parameter (f) at all.
"No (Medium)": the approach has partly satisfied the success parameter (f) better than "No (High)".

Future works: Despite its significant advantages, the primal-dual FOC approach poses intriguing challenges, addressed in subsequent chapters. Chapter 3 addresses the issue of *indirect* constraint control (d). Moreover, for highly interactive systems, the implementation is not as easy as weakly interactive ones. To address this issue, we recommend for primal-dual HRTO. Alternatively, Chapter 4 investigates strategies for applying primal-dual FOC to complex and/or large systems (g).

Chapter 3

Primal-Dual FOC with Override for RTO

To be able to employ direct constraint control in order to minimize critical constraint violations in primal-dual framework (that sill allowing automatic constraint switching), this chapter suggests integrating override controllers and auxiliary constraints in the framework. This chapter is based on the work in [56]

3.1 Introduction

As stated in Chapter 2, the primal dual FOC shows promising results, addressing the interesting issue of *indirectly* controlling constraints. This concern resonates with dual-based hybrid RTO, rendering strategies outlined in this chapter relevant for both approaches.

Before proceeding, it is important to have clear definition of *direct* and *indirect* constraint control as described in the following:

Definition 3.1: Direct Constraint Control

Direct constraint control is a technique of constraint control that controls a *measured* constrained variable (not through a constraint model) by *directly* updating a *primal* manipulated variables using a simple feedback controller such as a PID controller.

Definition 3.2: Indirect Constraint Control

Indirect constraint control is a technique of constraint control that controls a *measured* constrained variable (not through a constraint model) by *indirectly* updating a *primal* manipulated variables. Specifically, the measured constrained variable is controlled by a variable that is one of the parameters constructing another controlled variable that is controlled by updating a *primal* manipulated variables. For example, cascade constraint control.

From these definitions, we conclude that the reduced gradient approach offers *direct* constraints control, while the primal-dual approach utilize *indirect* constraints control. In

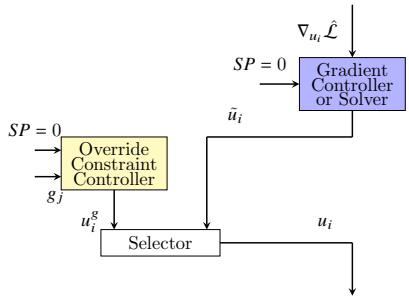


Figure 3.1: Incorporating override control for a constraint g_j

the primal-dual approach, a *measured* constrained variable is controlled by updating *dual* MVs, which is a parameter for the gradient of Lagrange function. To estimate the gradient of Lagrange function, we need *dual* MVs, steady-state cost and constraint gradient. Subsequently, *primal* MVs are updated to control this estimated gradient to zero. It is important to note that in this *indirect* control scheme, additional parameters can effect constraint control performance.

First, steady-state gradient estimation (both cost and constraint) relies on accurate states estimation and favorable matrix condition. To estimate states (and parameter/ disturbance), a well tuned (dynamic-) estimator and enough measurements are vital. Insufficient conditions in these calculations may lead to failure in constraint control, making *direct* constraint control indispensable.

Second, as the master controller lies on the outer loop, time scale separation must be considered for convergence. Consequently, constraint control operates at a slower time scale, resulting in slower updates of *dual* MVs and significant dynamic violations. Under conditions of large magnitude of constraint violation, having a hard constraint may incur economic losses due to necessary back-off applications. *Direct* constraints control offers tighter constraint control and reduce the back-off parameter.

Based on these *technical* motivations, we introduce the primal dual FOC with override.

3.2 Proposed Approach

3.2.1 Override Constraint Controllers

To control constraint g_j *directly* (and thus may reduce constraint violation and thus may minimize the need for back-off), we propose to pair it with a selected primal variable (input u_i) and control it using a (fast-) override controller. This is implemented at the bottom of the control hierarchy, as illustrated in Figure 3.1. Light blue box represents a gradient controller that is slower than the light yellow box (representing the override constraint controller). At any given time, the selector will choose the process input (u_i) as either the output from the override controller (u_i^g) or the output from the gradient controller or equation solver ($\tilde{u}_i$).

Choice of override pairings: It is important to find a good pairing between the constraint g_j and the selected input u_i:

$$u_i \leftrightarrow g_j$$

For choosing pairings and tuning purposes, one may consider the linear transfer matrix $\mathbf{G^g}(s)$ from the inputs to the override constraints:

$$\mathbf{g} = \mathbf{G^g}(s)\ \mathbf{u} \tag{3.1}$$

The following pairing rules are useful [46].

1. "Pair close rule": Select an input u_i with a large and direct effect on the constraint g_j [46]. For example, if the element $G^g_{ij}(s)$ from u_i to g_j is approximated as a first-order plus delay transfer function, then prefer a pairing with a large gain, a small delay, and a small time constant. One obvious way to obtain the relationship between the constraint g_j and the input u_i is by performing step response.
2. "Input saturation rule:" Select an input u_i that is not likely to saturate, for example, at a fully open or closed valve.

Note that we only need override control for constraints g_j where there are limitations on the allowed dynamic violations, for example, for hard constraints.

Tuning of override constraint controller: The name "override" is appropriate because we aim to make short-term corrections to the original steady-state optimization problem (2.1), with the goal of avoiding undesired dynamic constraint violations (see Figure 3.1). However, on a longer time scale, the steady-state optimization should take over and provide the optimal value of the input, $u_i = \tilde{u}_i$, and also decide whether on or not a constraint is active. This has implications for tuning, i.e., if the slower/outer layer is not slow enough, the closed-loop system may eventually lead to instability. To prevent the override controller from interfering with the steady-state optimization, we need a time scale separation between the fast override controller and its slower gradient controller, typically in the range 4 (minimum) to 10 (desired) [57].

Choice of override selector: Referring to the selector rule in [51], if constraint g_j has upper bound and the gain to the input u_i is negative or if constraint g_j has lower bound and the gain to the input u_i is positive (in other words, if the constraint g_j is satisfied by a large input u_i), we use max-selector:

$$u_i = \max\{\tilde{u}_i, u_i^g\}$$

Conversely, if constraint g_j has upper bound and the gain to the input u_i is positive or if constraint g_j has lower bound and the gain to the input u_i is negative (in other words, if the constraint g_j is satisfied by a small input u_i), then we use min-selector:

$$u_i = \min\{\tilde{u}_i, u_i^g\}$$

Maximum number of override constraints: In most cases, each override constraint is paired with a different input, meaning that the number of override constraints ($n_{g,o}$) cannot exceed the number of inputs, i.e., $n_{\mathbf{g},\mathbf{o}} \leq n_{\mathbf{u}}$. However, if two override constraints cannot be active simultaneously (for example, a variable with both an upper and lower bound), it may be possible to pair two override constraints with one input.

3.2.2 Auxiliary Constraints

Assigning both the master and override constraint controllers to control the constraint g_j may seem like a viable solution, but it fails to function as desired. The problem is that

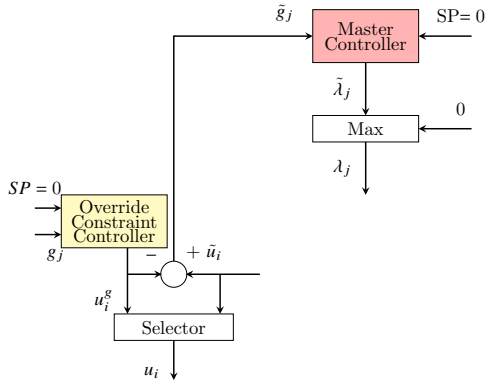


Figure 3.2: Controlling auxiliary constraint g_j using master controller

once the override takes over the constraint, the master controller will no longer update λ_i, and thus it will not release the constraint even when it no longer should be active.

To avoid this problem, we propose for override constraints to replace, in the master controller, the original constraint g_j with an auxiliary constraint $\tilde{g}_j$, which is the difference between the process input computed by the gradient controller and the process input computed by the override constraint controller, as expressed in Equation (3.2):

$$\tilde{g}_j = \tilde{u}_i - u_i^g \tag{3.2}$$

Figure 3.2 shows the proposed block diagram with override, incorporating the auxiliary constraint $\tilde{g}_j$. Light red box represents a master controller that is slower than the light yellow box (representing the override constraint controller). For constraints g_j with override control, the master controller is assigned to control the auxiliary constraint $\tilde{g}_j = \tilde{u}_i - u_i^g$. The sign of the gain in the master controller depends on whether it is a "`min`" or "`max`" selector at the bottom of the hierarchy. The complete block diagram, which includes the gradient controller, is shown in Figure 3.3.

For the master controller, the auxiliary constraint is $\tilde{g}_j < 0$ if the constraint g_j is satisfied by a small input u_i (and thus we have a min-selector for the override), and $-\tilde{g}_j < 0$ if the constraint g_j is satisfied by a large input u_i (and thus we have a max-selector for the override). To better understand the use of auxiliary constraints, we consider the two directions of constraint switching:

- If a constraint g_j is originally *not* active and a disturbance causes g_j to be violated, the override controller will change u_i^g until the selector assigns $u_i = u_i^g$. At this point, we get a violation (nonzero value) of the auxiliary constraint, and the master controller will increase (slowly) the associated dual variable λ_j (which is zero in the unconstrained case), until we achieve $\tilde{g}_j = 0$, where the override and gradient controllers agree on the value for the process input.
- If a constraint g_j is originally active, and a disturbance causes this to no longer be optimal, the master controller will decrease λ_j, which again changes $u_i = \tilde{u}_i$ (because the override controller is no longer active), until we get $\lambda_j = 0$ where the constraint is no longer controlled.

Remark 3.1: Introducing Master Controllers

In Chapter 2, the outer layer controller, termed constraint controller in the primal-dual framework, is responsible for both constraint control and active constraint switching during transient. In our proposed approach, this controller, now called the master controller, solely handles active constraint switching during transient, while a separate override constraint controller focuses on constraint control during transient.

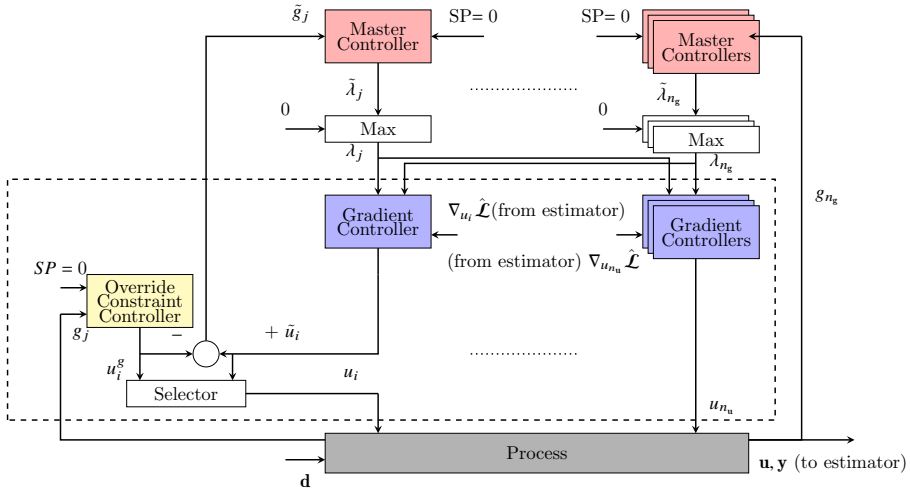


Figure 3.3: Complete structure of the proposed scheme with gradient controllers and one override controller with its associated auxiliary constraint $\tilde{g}_j$.

3.2.3 Implicit override for input constraints

All physical process inputs have upper and lower constraints (also know as saturation limits)

$$u_i \leq u_i^{max} \qquad\qquad u_i \geq u_i^{min}$$

or equivalently

$$g_j^{max} = u_i - u_i^{max} \leq 0 \tag{3.3a}$$
$$g_j^{min} = u_i^{min} - u_i \leq 0 \tag{3.3b}$$

These input constraints are always "hard" because they cannot be physically violated. Thus, physics provide an implicit override, and an override controller is not necessary for input constraints. However, we still need to use the auxiliary constraint for the master controller, as shown in Figure 3.4. Note that physics provide an implicit override, so the grey Min-selector at the bottom is not part of the control system, but represents the physical input saturation (valve).

$$\tilde{g}_j^{max} = \tilde{u}_i - u_i^{max} \leq 0 \tag{3.4a}$$
$$\tilde{g}_j^{min} = u_i^{min} - \tilde{u}_i \leq 0 \tag{3.4b}$$

If the master controller instead were to control the (physical) constraint (u_i^{max} or u_i^{min}), then it would not be possible to get out of the constraint when it is no longer optimally active.

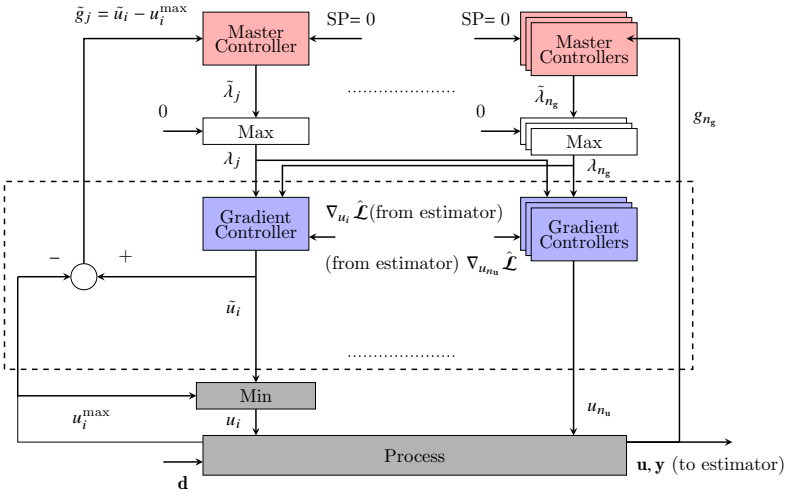


Figure 3.4: Proposed use of auxiliary constraint for case with maximum input constraint on u_i.

3.3 Numerical Examples

This section presents a case study to illustrate the effectiveness of the primal-dual feedback-optimizing control structure, both with and without override constraint control. The case study involves a gas-lift oil production system with four oil production wells and a riser, as depicted in Figure 3.5. The main objective is to maximize the total oil production (q_o). Local input constraints exist, specifically on the maximum valve openings for the four gas lift choke valves, along with a coupling constraint on the downstream gas handling capacity (q_g).

The system includes a gas-lift distribution network with four gas lift valves (MVs), a manifold, and a riser which gives the export gas and produced (export) oil. This system is a variant of a system that has been studied and used in [41, 58, 59, 60]. The model description is available in Appendix B, and its parameters are shown in Table 3.1.

3.3.1 Case Description

The system is described by a set of coupled differential-algebraic equations (DAEs). There are four manipulated variables (primal variables, inputs), which are the openings of the four gas-lift valves:

$$\mathbf{u} = [u_1 \ \dots \ u_4]^\top$$

The system is weakly interactive because a change in one gas lift valve opening impacts the gas lift and oil and gas flow rates of all wells. The considered disturbances are the gas-oil ratio in the inflow from the reservoir for wells 2. Furthermore, since the downstream process may change over the time, we also consider the maximum gas export q_g^{max} as

Table 3.1: Parameter used in Matlab model

Annulus				
Parameter	1	2	3	4
p_s [Bar]	200			
r_a [m]	0.2032	0.2032	0.2032	0.2032
L_a [m]	2400	2400	2400	2400
T_a [K]	318	318	318	318
c_{ga} [m^3/(s.Bar)]	5.52e-5	5.58e-5	5.46e-5	5.73e-5
L_j [m]	1500	1500	1500	1500
c_{gl} [m^3/(s.Bar)]	1.52e-5	1.66e-5	3.13e-5	3.53e-5
Well tubings				
Parameter	1	2	3	4
r_t [m]	0.091	0.091	0.091	0.091
L_t [m]	2400	2400	2400	2400
T_t [K]	323	323	323	323
c_{fh} [m^3/(s.Bar)]	8.12e-5	7.86e-5	8.37e-5	8.77e-5
w_t [-]	0.1	0.1	0.1	0.1
Q [m^3/s]	0.0347	0.0347	0.0347	0.0347
p_r [Bar]	219	218	220	216
n [-]	0.8	0.8	0.8	0.8
ρ_{ot} [kg/m^3]	800	800	800	800
ρ_{gt} [kg/m^3]	20	20	20	20
Riser				
Parameter				
r_e [m]	0.6096			
L_e [m]	250			
T_e [K]	313			
c_{fe} [m^3/(s.Bar)]	0.0010			
w_e [m]	0.1			
p_d [Bar]	15			
Common				
Parameter				
k [m^2/s]	9.81			
R [J/(K mol)]	8.314			
M_g [kg/mol]	0.020			
p_o [$]	503.20			
p_l [$]	0.25			

disturbance.

$$\mathbf{d} = \begin{bmatrix} GOR_2 & q_g^{max} \end{bmatrix}^\top$$

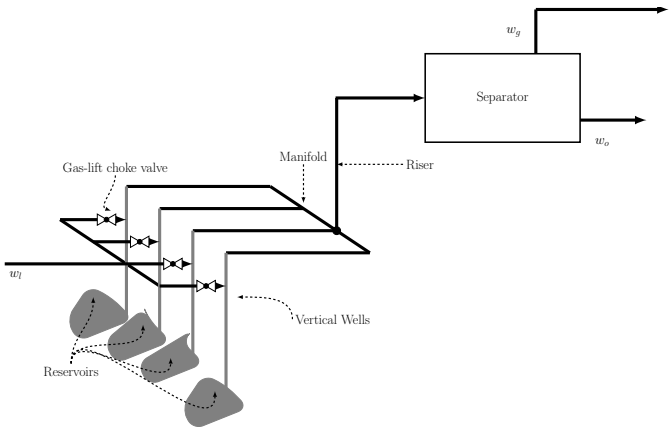


Figure 3.5: Flowsheet for gas lift case study.

The main objective is to maximize the total oil production q_o. At steady state we have

$$q_o = q_{o,1} + q_{o,2} + q_{o,3} + q_{o,4}$$

where $q_{o,i}$ is the produced oil from reservoir i. However, at the same time, one should try to minimize the total gas lift supply,

$$q_l = q_{l,1} + q_{l,2} + q_{l,3} + q_{l,4}$$

The maximum constraint on the downstream processing of the total export gas, q_g provides a coupling constraint. At steady state, we have

$$q_g = q_{g,1} + q_{g,2} + q_{g,3} + q_{g,4} + q_l$$

where $q_{g,i}$ is the produced gas from reservoir i.

Any excess gas is flared (burned), which is strongly undesirable for environmental reasons. Therefore, override control is used for the coupling constraint (denoted g_1 below). In addition to the coupling constraint, each gas-lift choke u_i has a physical constraint with a maximum valve opening of 1. In summary, the steady-state optimization problem can be expressed as follows:

$$\min_{\mathbf{u}} \quad J = -p_o q_o + p_l q_l \tag{3.5a}$$

$$\text{s.t.} \quad g_1 = q_g - q_g^{max} \leq 0, \tag{3.5b}$$

$$g^{u_i^{\max}} = u_i - u_i^{max} \leq 0,\ i = 1, \ldots, 4 \tag{3.5c}$$

Here, p_o is the price of oil, p_l is the cost of gas lift supply, q_g^{max} is the maximum gas export, and $u_i^{max} = 1$ is the maximum opening of gas lift valve i.

3.3.2 Active constraint regions

For this particular problem, we have five constraints, $n_{\mathbf{g}} = 5$ (one coupling constraint and four max-constraints on the inputs), and therefore a maximum of $2^{n_{\mathbf{g}}} = 32$ active constraint regions is possible. However, only a subset of these regions are encountered in practice. To illustrate this, consider a scenario where q_g^{max} varies between 0.0085 and 0.0150 m^3/s (equivalent to between 734.40 and 1296.00 m^3/day) and GOR_2 varies between 0.01 and 0.15 m^3/m^3, while the other GOR_i (for $i = 1, 3, 4$) are constant. For these two disturbances, Figure 3.6 shows the 4 possible active constraint regions for optimal operation. Given constant GOR_1, GOR_3, and GOR_4, the four possible active constraint regions are labeled R-I to R-IV: R-I is a region where g_1, g_2, and g_3 are active, R-II is a region where g_1 and g_2 are active, R-III is a region where g_1 and g_3 are active, and R-IV is a region where only the coupling constraint g_1 (max. gas handling capacity) is active.

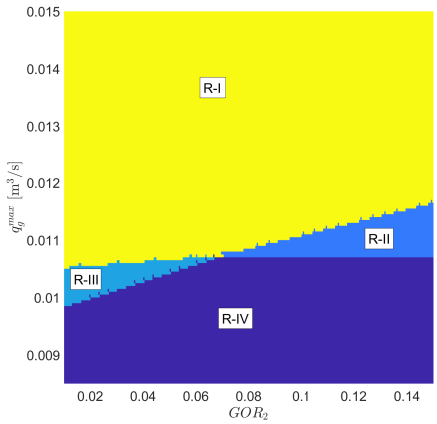


Figure 3.6: Optimal active constraint regions as a function of the two disturbances, q_g^{max}, and GOR_2.

Actually, for our purposes the details are not important, because it does not matter for our proposed method how many regions we may encounter and which constraints transitions may occur. This is because the primal-dual feedback optimizing scheme can handle any number of regions and transitions between them. However, to guarantee that we can optimally implement override control for g_1, we must require that $n_u \geq n_{g,o}$, where $n_{g,o}$ is the number of override constraints. This condition is not satisfied in our case since the four maximum gas-lift valve openings are implicitly hard constraints, so we have $n_u = 4$ and $n_{g,o} = 5$. Fortunately, this is not a problem for our case study because two of the input constraints (on inputs 3 and 4) are never active for the assumed disturbances.

In these simulations, we consider the two disturbances, GOR_2, and q_g^{max} and they are varied over time. At $t = 5$ hr, GOR_2 gradually decreases in 5 minutes from 0.1200 to 0.0360, rebounding to 0.1240 in 5 minutes at $t = 21$ hr. Meanwhile, q_g^{max} gradually decreases in 5 minutes from 933.12 to 743.04 m^3/day at $t = 1$ hr, gradually increases in 5 minutes to 915.84 m^3/day at $t = 5$ hr, further gradually increases in 5 minutes to 1114.56 m^3/day at $t = 13$ hr, and finally drops in 5 minutes to 864.00 m^3/day at $t = 21$ hr.

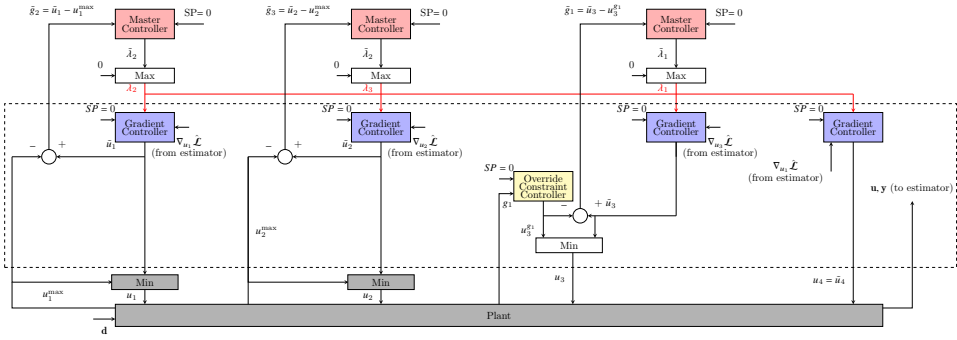


Figure 3.7: Proposed primal-dual control scheme for the case study.

3.3.3 Simplified problem description

Since $n_g = 5$ and $n_u = 4$, the primal-dual feedback-optimizing control ideally requires 5 master controllers and 4 gradient controllers. However, in the considered disturbance scenario (Figure 3.6), constraints $g^{u_3^{\max}}$ and $g^{u_4^{\max}}$ are never active, implying that their associated Lagrange multipliers λ_i are always zero. Hence, the number of required master controllers can be reduced to three. In the remaining sections of the paper, we consider the following three constraints:

$$\mathbf{g} = \begin{bmatrix} g_1 \\ g_2 \\ g_3 \end{bmatrix} = \begin{bmatrix} g_1 \\ g^{u_1^{\max}} \\ g^{u_2^{\max}} \end{bmatrix} = \begin{bmatrix} q_g - q_g^{max} \\ u_1 - u_1^{\max} \\ u_2 - u_2^{\max} \end{bmatrix} \tag{3.6}$$

The associated Lagrange multipliers are denoted λ_1, λ_2, and λ_3, respectively.

The proposed control structure is shown in Figure 3.7. For the case without override control, $\tilde{g}_1$ is replaced by the measured constraint g_1. The red lines indicate that all three Lagrange multipliers are distributed to each of the four gradient controllers. All controllers are SIMC (Simple Internal Model Control)-tuned PI-controllers [52], with the closed-loop time constant τ_c as the tuning parameter. Note that we need a time scale separation between the fast override controller in the lower layer, the intermediate gradient controllers in the middle layer and the slower master constraint controllers in the upper layer. Typically, we need a time scale separation of 5 or more between each layer.

In the simulations, we will consider two cases: one with the override controller on the coupling constraint g_1 and one without.

3.3.4 Gradient controllers

For tuning the gradient controllers, we consider the square steady-state gain matrix G from the inputs (u) to the gradients ($\nabla_u \mathcal{L}$). Linearizing in region R-IV gives:

$$\mathbf{G} = \begin{bmatrix} 0.4092 & -0.0069 & -0.0139 & -0.0159 \\ -0.0068 & 0.5106 & -0.0150 & -0.0171 \\ -0.0131 & -0.0142 & 1.8861 & -0.0327 \\ -0.0149 & -0.0161 & -0.0325 & 2.3963 \end{bmatrix} \tag{3.7}$$

Table 3.2: Controller and Tuning parameters-1

Description	Variable	Value
Dual-based without override on the coupling constraint		
Master Controller 1 Proportional Gain	$K_{P,1}$	-1.2833e+04
Master Controller 1 Integral Gain	$K_{I,1}$	-57.0362
Master Controller 1 Anti Windup Gain	$K_{AW,1}$	2.2500
Master Controller 2 Proportional Gain	$K_{P,2}$	-0.0168
Master Controller 2 Integral Gain	$K_{I,2}$	-0.0010
Master Controller 2 Proportional Gain (when constraint active)	$K_{P,2}$	-0.3095
Master Controller 2 Integral Gain (when constraint active)	$K_{I,2}$	-0.0193
Master Controller 2 Anti Windup Gain	$K_{AW,2}$	0.1600
Master Controller 3 Proportional Gain	$K_{P,3}$	-0.0114
Master Controller 3 Integral Gain	$K_{I,3}$	-0.0006
Master Controller 3 Proportional Gain (when constraint active)	$K_{P,3}$	-0.3172
Master Controller 3 Integral Gain (when constraint active)	$K_{I,3}$	-0.0176
Master Controller 3 Anti Windup Gain	$K_{AW,3}$	1.8000e-05
Gradient Controller 1 Proportional Gain	$K_{P,1}$	1.0740
Gradient Controller 1 Integral Gain	$K_{I,1}$	0.0268
Gradient Controller 1 Anti Windup Gain	$K_{AW,1}$	0.4000
Gradient Controller 2 Proportional Gain	$K_{P,2}$	0.8823
Gradient Controller 2 Integral Gain	$K_{I,2}$	0.0215
Gradient Controller 2 Anti Windup Gain	$K_{AW,2}$	0.4100
Gradient Controller 3 Proportional Gain	$K_{P,3}$	0.2331
Gradient Controller 3 Integral Gain	$K_{I,3}$	0.0058
Gradient Controller 4 Proportional Gain	$K_{P,4}$	0.1788
Gradient Controller 4 Integral Gain	$K_{I,4}$	0.0046

We notice that there are some interactions between the wells, as expected. However, the interactions are small, and single-loop (decentralized) gradient controllers will work well.

The four gradient controllers were tuned with a closed-loop time constant (τ_c) of 1.5 minutes. The resulting controllers tuning are shown in Table 3.2-3.3.

3.3.5 Master constraint controllers

With the four lower-layer gradient controllers tuned and functioning properly, we tune the three master constraint controllers. Note that the coupling constraint g_1 is a common constraint, whereas $g_2 = g^{u_1^{max}}$ and $g_3 = g^{u_2^{max}}$ represent local input constraints.

The three master constraint controllers were tuned with a closed-loop time constant (τ_c) of 7.5 minutes, corresponding to a time scale separation of 5 relative to the four gradient controllers. The resulting controllers tuning are shown in Table 3.2-3.3.

Table 3.3: Controller and Tuning parameters-2

Description	Variable	Value
Dual-based with override on the coupling constraint		
Master Controller 1 Integral Gain	$K_{I,1}$	-104.1411
Master Controller 1 Anti Windup Gain	$K_{AW,1}$	1.0000e-04
Master Controller 2 Proportional Gain	$K_{P,2}$	-0.0168
Master Controller 2 Integral Gain	$K_{I,2}$	-0.0010
Master Controller 2 Proportional Gain (when constraint active)	$K_{P,2}$	-0.3095
Master Controller 2 Integral Gain (when constraint active)	$K_{I,2}$	-0.0193
Master Controller 2 Anti Windup Gain	$K_{AW,2}$	0.1600
Master Controller 3 Proportional Gain	$K_{P,3}$	-0.0114
Master Controller 3 Integral Gain	$K_{I,3}$	-0.0006
Master Controller 3 Proportional Gain (when constraint active)	$K_{P,3}$	-0.3172
Master Controller 3 Integral Gain (when constraint active)	$K_{I,3}$	-0.0176
Master Controller 3 Anti Windup Gain	$K_{AW,3}$	1.8000e-05
Gradient Controller 1 Proportional Gain	$K_{P,1}$	1.0740
Gradient Controller 1 Integral Gain	$K_{I,1}$	0.0268
Gradient Controller 1 Anti Windup Gain	$K_{AW,1}$	0.4000
Gradient Controller 2 Proportional Gain	$K_{P,2}$	0.8823
Gradient Controller 2 Integral Gain	$K_{I,2}$	0.0215
Gradient Controller 2 Anti Windup Gain	$K_{AW,2}$	0.4100
Gradient Controller 3 Proportional Gain	$K_{P,3}$	0.2331
Gradient Controller 3 Integral Gain	$K_{I,3}$	0.0058
Gradient Controller 3 Anti Windup Gain	$K_{AW,3}$	0.0040
Gradient Controller 4 Proportional Gain	$K_{P,4}$	0.1788
Gradient Controller 4 Integral Gain	$K_{I,4}$	0.0046
Override Controller 3 Proportional Gain	$K_{P,3}$	722.9696
Override Controller 3 Integral Gain	$K_{I,3}$	13.6409
Override Controller 3 Anti Windup Gain	$K_{AW,3}$	0.0053

3.3.6 Override constraint controller

Inputs u_3 and u_4 never saturate and are therefore candidates for override control for the coupling constraint g_1. For the case study, we chose to use u_3. Inputs u_1 and u_2 may saturate, but the physical valve provides indirect override for these inputs.

The override controller for g_1 was tuned with a closed-loop time constant (τ_c) of 0.3 minutes, which again gives a time scale separation of 5. The resulting controllers tuning are shown in Table 3.3.

3.3.7 Gradient estimator

To evaluate the gradient of the Lagrange function with respect to the inputs, it is necessary to estimate the steady-state cost gradient and steady-state constraint gradient. In this

case study, we follow [55] and linearize at each sample time the nonlinear model to obtain a linear model from u to the states x,

$$\dot{\mathbf{x}} = \mathbf{A}\boldsymbol{\Delta}\mathbf{x} + \mathbf{B}\boldsymbol{\Delta}\mathbf{u}$$

(in deviation variables). The static model for the cost $J(x, u, d)$ is linearized in a similar way to obtain

$$\Delta J = \mathbf{C}\boldsymbol{\Delta}\mathbf{x} + \mathbf{D}\boldsymbol{\Delta}\mathbf{u}$$

We do not need to include the dependency of $\dot{\mathbf{x}}$ and J on the disturbances d, because the disturbances are assumed constant into the future ($\Delta d = 0$), when we follow the "trick" in [55] of setting $\dot{\mathbf{x}} = \mathbf{0}$ to eliminate Δx and obtain $\Delta J = \nabla_{\mathbf{u}}\mathbf{J}\Delta u$. The estimate of the steady-state cost gradient then becomes

$$\nabla_{\mathbf{u}}\mathbf{J} = -\mathbf{C}\mathbf{A}^{-\mathbf{1}}\mathbf{B} + \mathbf{D} \tag{3.8}$$

In this paper, we use an extended Kalman filter (EKF) to estimate the states and disturbances and based on these estimates, we relinearize the system at each sample time to obtain the matrices A, B, C and D.

The same approach is used to estimate the constraint gradient $\nabla_{u_i}\mathbf{g}_j\,(\mathbf{u}, \mathbf{d})$.

These estimates can be achieved using any model-based or model-free gradient estimation. If unmeasurable state variables are present in model-based estimation, a novel piecewise fuzzy affine observer was introduced. See [61] for details. For various gradient estimation techniques for RTO, refer to [35].

3.3.8 Simulation Results for *Tight* Constraint Control

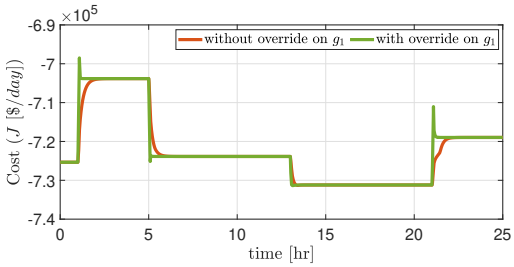
We present simulation results for the proposed primal-dual structure in Figure 3.7, both with and without override control. Figure 3.8(a) shows that we reach the steady-state optimal operating points both with and without override. Figure 3.8(b) shows that the override gives much better control of the coupling constraint (g_1). Table 3.4 shows this in more detail. We note that the amount of flared gas is reduced by about a factor 67, and the maximum violation of the constraint is reduced by about a factor 17. Figure 3.9 shows the four inputs (valve positions) for the case with override. It shows that the proposed approach is able to move correctly to all the active constraints.

Table 3.4: Violation of constraint g_1 for case study

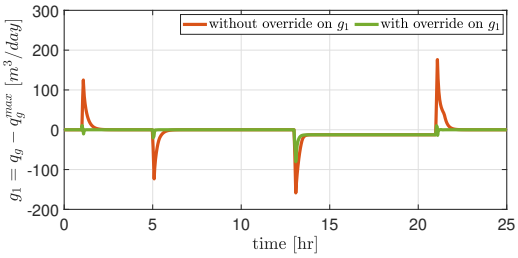
	Without override	With override
Max. constr. violation [m^3/day]	176.2760	10.6436
Flared gas (int. violation) [m^3]	2.9667	0.0444
Average flared gas [m^3/hr]	0.1187	0.0018

3.3.9 Numerical Demonstration for Back-off Reduction

For the present case study, we assume that the small amount of flared gas (0.0444 m^3 in 25 hour) obtained with the override is acceptable, so we introduce a back-off on the case without override to get a similar constraint violation (flaring). We find that the required



(a) Cost J [$/day].



(b) Coupling constraint g_1.

Figure 3.8: Simulation results with override (green) and without override (red). Without override we get constraint violation with $g_1 > 0$.

back-off on the export gas is $b_1 = 137.9359$ m^3/day. In practice, this is implemented by changing the setpoint for g_1 from SP=0 to SP=$-b_1 = -137.9359$.

The resulting economic loss by adding this back-off is shown in Figure 3.10 where we plot as a function of time the accumulated loss (cost difference), $\Delta J = J_{back-off} - J_{override}$.

Figure 3.11 shows the value of the Lagrange multiplier λ_1 for the coupling constraint g_1 as a function of time, in which the magnitude is correlated with the slope of economic loss, shown in Figure 3.10.

Recall that the loss imposed by back-off is equal to $|\lambda_1 b_1|$ [$/s]. In our case, we have $|b_1| = 137.9359$ m^3/day $= 0.0016$m^3/s and from Figure 3.11 (yellow line), we find that λ_1 is 125.1683 [$/m^3] on average. The average loss is then approximately $|\lambda_1 b_1| = 0.2003$ [$/s] and the accumulated loss over 25 hours is then $0.2003 \cdot 25 \cdot 3600 \approx 18000$ $ which quite agrees well with the final value of about 14000 $ in Figure 3.10. The 4000 gap discrepancy can be attributed to the substantial length of the unconstrained case during the time window between $t = 13$ hr and $t = 21$ hr.

Furthermore, the ideal average profit is $722,275$/day, as can be observed in Figure 3.8(a), which means that implementing override can save approximately 2% compared to "back-off" strategy.

3.3.10 Numerical Demonstration for Auxiliary input constraints

To emphasize the need to use the auxiliary input constraint (3.4) in the master controller (green lines in simulations), rather than the actual input constraint (3.3) (purple lines), we

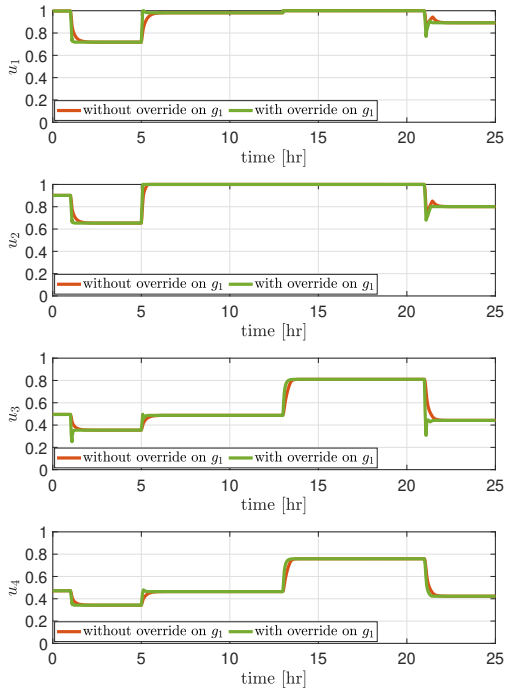


Figure 3.9: Simulation results with override (green) and without override (red). Inputs u_i (gas lift valve positions).

compare the gradients for the two cases in Figure 3.12. We see that without the auxiliary constraint (purple line), input u_1 and u_2 (see Figure 3.13) do not violate their constraints and therefore, there is no update of the corresponding Lagrange multiplier, and the static optimality condition is not satisfied.

The corresponding Lagrange multipliers for the two cases are shown in Figure 3.14. Here, we see more clearly that the inputs become constrained (with $\lambda_2 > 0$ or $\lambda_3 > 0$) for some period with the auxiliary constraint (green lines).

3.4 Discussion

3.4.1 Comparison with Dynamic RTO or ENMPC

One obvious solution to reduce dynamic violations involves employing dynamic optimization tools such as dynamic RTO or economic nonlinear model predictive control (ENMPC). Yet, these tools demand high computational capacity, which is challenging even with modern computing capabilities. Although extensively used in research papers, recent research [62, 25] highlights several numerical challenges hindering widespread adoption of dynamic optimization. Additionally, safety requirements in many process industries may require deploying automatic tools on embedded platforms such as programmable logic controllers (PLC), which is currently unsuitable for solving nonlinear optimization online [63].

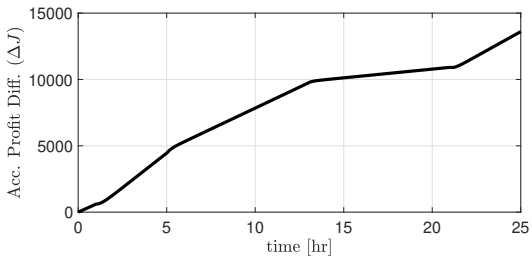


Figure 3.10: Accumulated economic loss [$] caused by using "back-off" for case without override (the back-off is chosen to achieve the same constraint violation as with override).

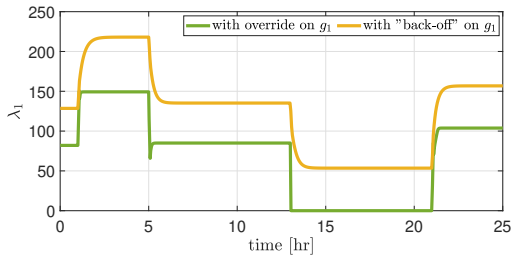


Figure 3.11: The Lagrange multiplier λ_1 with "back-off" (and without override) (yellow line) is seen to be larger than with override (green line).

In general, compared to the standard numerical solver-based RTO, the proposed approach offers a lower level of complexity, and the constraints are measured and controlled in a *transparent* manner (without relying on a model for the constraints) in the upper layer. Additionally, when obtaining stationary condition using feedback controllers, the computation time requirements are much less, as it uses only PID controllers.

3.4.2 Override as *Direct* Constraint Control

As *direct* constraint control, the override controller anticipates the effect of inaccurate additional/intermediate parameters, i.e., steady-state gradient estimation, in the proposed structure. This feature does not exist in the original primal-dual framework.

3.4.3 Comparison with Method of Multipliers

The method of multipliers (MoM) or augmented Lagrangian [64] involves incorporating penalty terms in the Lagrangian cost $\mathcal{L}$ to enforce constraint satisfaction during the (numerical) minimization of $\mathcal{L}$.

Consequently, in the implementation of MoM (discretized representation), the master controller (equipped with a `max` selector and weight $(\boldsymbol{\rho})$ on the constraint) is however limited by the time scale separation concept [65, 46], rendering the penalty terms insignificant if the constraint is controlled in a slow time scale.

$$\boldsymbol{\lambda}^{k+1} = \max\left[0, \boldsymbol{\lambda}^{k} + \boldsymbol{\rho}\mathbf{g}\right] \tag{3.9}$$

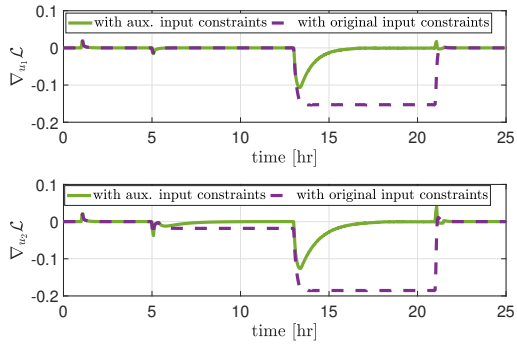


Figure 3.12: Gradients of Lagrange function with respect to inputs u_1 and u_2 with override control. The gradients should optimally go to zero at steady state, which is achieved when we use the auxiliary input constraints (3.4) (green lines) but not when we incorrectly use the original input constraints (3.3) in the master controller (purple lines).

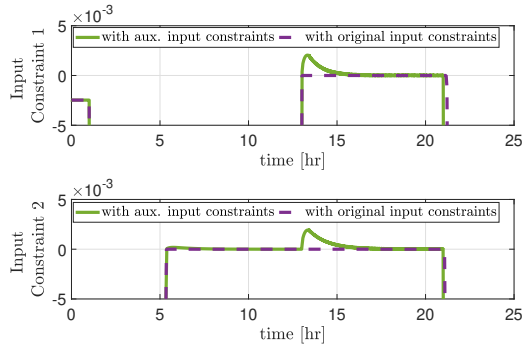


Figure 3.13: Input constraints with override control. The use of auxiliary input constraints (3.4) (green lines) allows for $\tilde{u}$ to temporary violate the constraint on u so that the corresponding Lagrange multiplier can get updated. Without the auxiliary constraint, this is not possible (purple lines).

Even with an appropriate (and enforced) value of $\boldsymbol{\rho}$, for the input constraint case, incorporating "back-off" remains necessary to minimize constraint violation further. This suggests that the performance of this strategy may not surpass that of using override and "auxiliary" constraints. This outcome is expected because the selector in the proposed strategy immediately eliminates the term from the "upper layer" in the calculation of the controlled variables ($\mathbf{c}(\boldsymbol{\lambda}) = \mathbf{g}(\mathbf{u}, \mathbf{y})$).

3.5 Chapter Summary

In this chapter, we proposed primal-dual FOC with override and auxiliary constraints to address the issue of *indirect* constraint control. This improvement not only enables *direct* constraint control but also enable *tight* constraint control (on fast time scale), and thus minimize constraint violations. For hard constraints, this approach may reduce constraint violation magnitudes and minimize back-off parameters. Denoting S-VII as dual-based

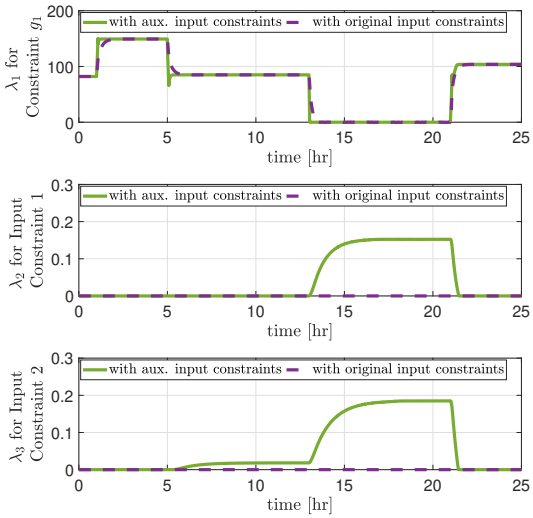


Figure 3.14: Lagrange multipliers with override control with and without auxiliary input constraints.

HRTO with override control and S-VIII as primal-dual FOC with override, we evaluate and compare parameters outlined in section 2.5. Table 3.5 summarizes the assessment, where the approaches are:

- S-IV: Dual-based hybrid RTO
- S-VI: Primal dual FOC
- S-VII: Dual-based HRTO with override control
- S-VIII: Primal-dual FOC with override

and the parameters we assess and compare are:

- (a): Is steady-state optimal achieved?
- (b): Is constraint controlled *transparently* (see Definition 2.1)?
- (c): Is flexible for active constraint switching?
- (d): Is constraint controlled *directly* (on fast time scale)?
- (e): Is applicable for less than twice differentiable Lagrange function?
- (f): Is numerical solver avoidable (computation time)?
- (g): Is it recommended for complex and large system?
- (h): Is input filter (or additional setpoint controller) avoidable?
- (i): Is parameters and states dynamic estimator avoidable?
- (j): Is gradient estimator avoidable?

Future works: In Chapter 2, primal-dual FOC with and without override are less general for highly interactive systems, posing implementation challenges compared to weakly interactive systems. To address this, we propose employing dual-based hybrid RTO with override, albeit with limitations concerning Lagrange function type, computation time, and the procedure for input filter (or additional setpoint controllers). These limitations are absent in the primal-dual FOC with override approach. Hence, Chapter 4 explores strategies for applying primal-dual FOC in complex and large systems (g) to resolve these issues.

Table 3.5: Summary of Chapter 3

	S-IV	S-VI	S-VII	S-VIII
(a)	Yes	Yes	Yes	Yes
(b)	Yes	Yes	Yes	Yes
(c)	Yes	Yes	Yes	Yes
(d)	No[1]	No[1]	Yes	Yes
(e)	No	Yes	No	Yes
(f)	No (Medium)	Yes	No (Medium)	Yes
(g)	Yes	No[2]	Yes	No[2]
(h)	No	Yes	No	Yes
(i)	No	No	No	No
(j)	Yes	No	Yes	No

"Yes": the approach has satisfied the success parameter
"Yes": the approach has satisfied the success parameter in this Chapter
"No": the approach does not satisfied the success parameter
"No": the approach does not satisfied the success parameter, and will be addressed in this thesis.
[1]: is addressed in Chapter 3
[2]: will be addressed in Chapter 4
"No (High)": the approach does not satisfied the success parameter (f) at all (shown in Chapter 2).
"No (Medium)": the approach has partly satisfied the success parameter (f) better than "No (High)".

Part II: Distributed Feedback-optimizing System (For Large-scale System)

Chapter 4

Dual-based DFoS Framework: From Interactive to Non-interactive

This chapter suggests and describes a Dual-based Distributed Feedback Optimization System framework. This framework is an efficient solution that employs simple tools to guide a decomposed large-scale process system towards optimal operation while minimizing information sharing. Additionally, this framework handles varying time scale separations among subprocesses and avoids numerical robustness issues by reducing reliance on optimization solvers. This chapter is based on the work in [41]

4.1 Introduction

In the process and manufacturing industry, coordinating several operating units, also known as a *coordinated large-scale process* presents a critical challenge. Different operational group, each responsible for distinct units and local optimization objectives, collaborate by sharing resources to achieve mutual benefit. To optimize the operations, real-time decision-making is essential to meet production targets and emission standards, typically facilitated by real-time optimization (RTO) utilizing process models and real-time measurements.

However, as industries strive for optimal coordination, they encounter new challenges. Efficiency drives them to minimize information sharing, necessitating effective resource allocation with minimal information exchange.

A promising solution to this challenge involves distributed real-time optimization, using the dual decomposition approach[66]. Here, individual subsystems, representing different entities, perform local optimization. Simultaneously, a master coordinator (*equivalent to a master controller in primal-dual, see Chapter 2*) adjusts shadow prices (Lagrange multipliers) for shared constraints. This minimizes the requirement for extensive information exchange, as only shadow prices and constraints are shared between subsystems and the master controller.

The master controller adjusts shadow prices based on the *concept of supply-demand dynamics*, encouraging increased consumption by subsystems when supply exceeds demand and reducing consumption when demand surpasses supply. This coordination mimics the tâtonnement process, wherein the controller iteratively finding equilibrium prices based on subsystem responses [38, 67, 64, 66]. To obtain a feasible and optimal solution, subproblems with local subsystem models and the master controller iterate within an optimization's numerical solver. This illustrates the core principles of this framework and its role in optimizing processes with limited information sharing.

Generally, distributed optimization comprises two main schemes: primal-based and dual-based. This chapter primarily focuses on dual-based optimization, addressing two key research gaps within this framework.

First research gap and suggested solution: The dual-based optimization method, outlined in [13], assumes an additive separable cost function, but is primarily suited for non-interactive systems, as per Definition 2.4. However, real-world scenarios often involve interactive systems. Hence, the first goal is to transform interactive systems into non-interactive ones.

This chapter suggests strategies for enabling transformation, including *identifying locally* predicted solution variables, *reformulating* problems, establishing coordinators for achieving *consensus* of predicted solutions among subsystems, and implementing solution predictors. These strategies create a non-interactive system, enabling subsystems to estimate disturbances and parameters locally, such as the gradient of the Lagrange function.

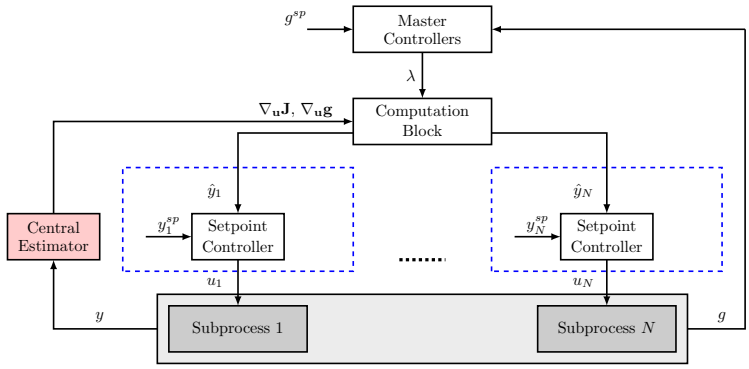
Second research gap and suggested solution: Typically, industries often favor simple feedback controllers over numerical solver due to limited expertise and a corporate culture prioritizing simplicity. A promising solution lies in *feedback-optimizing control*, as advocated by [28], which translates optimization directly into the control layer, eliminating the need for solving numerical optimization problems.

Feedback optimizing control, extensively studied for single optimization problems, focuses primarily on what to control for the unconstrained degrees of freedom. Key references include [8, 68, 69], and [70]. Other approaches like NCO-tracking, hill-climbing, extremum seeking, and feedback RTO aim to drive the estimated steady-state cost gradient to zero [71, 12, 72, 73]. However, limited research exists on feedback optimizing control in distributed RTO with a master coordinator, which is also a gap addressed in this Chapter.

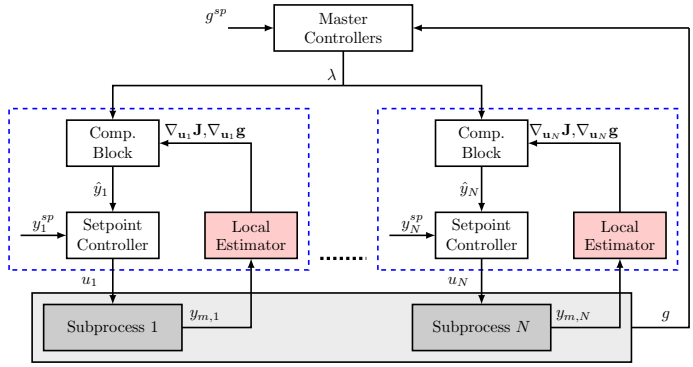
To eliminates the necessity of solving numerical optimization problems, this chapter introduces a self-optimizing controlled variable for each subproblem, which is given as a function of the shadow price. By independently controlling these proposed local controlled variables, individual units can locally optimize their processes based on the current shadow price. Through continuous updates of shadow prices by the master controller, optimal system-wide operation is attained.

In this thesis, we introduce the framework as the dual-based *distributed feedback-optimizing system* (dFoS). To distinguish between dFoS and *distributed feedback-optimizing control* (dFoC), please refer to Section 1.2 and Figure 1.5(b)

To address the first research gap, we utilize a local self-optimizing controller for a solution prediction, while the master controller acts as a consensus coordinator, distinguishing the dual-based dFoS from the primal-dual FoC by enabling subsystem decomposition into several independent subsystems.



(a) Dual-based distributed feedback-optimizing control (dFoC), also known as primal dual FoC.



(b) Dual-based distributed feedback-optimizing system (dFoS)

Figure 4.1: Graphical illustration showcasing the distinction between of *Primal-dual FOC* and Dual-based *dFoS*

The main contribution of this chapter is the introduction of a dual-based dFoS framework for interactive systems. This enables optimal steady-state operation in a distributed manner with limited information sharing and utilizing simple feedback controllers. This approach addresses numerical issues, enhances transparency for easier implementation, maintenance, and tuning by plant personnel, and crucially enables time scale separation between subproblems and master controllers. The manipulated variables from the subproblems can be updated on a faster time scale than the master controllers, allowing for different tunings, closed-loop time constants, and sampling intervals for individual subproblems. This may be a significant advantage in many large-scale industrial processes, for example, because the process dynamics, measurement delays, etc., may vary for different subprocesses.

To provide clarity, Figure 4.1 serves to illustrate the distinction between dual-based *distributed feedback-optimizing control* (discussed in Chapter 2, represented in Figure 4.1(a)) and the dual-based *distributed feedback-optimizing system* (dFoS) approach (Figure 4.1(b)).

4.2 Problem Formulation

For clarity, consider a steady-state optimization problem with two (2) distinct subsystems or units. The problem can be defined as follows:

$$\min_{u_1,\,u_2} \quad J\left(u_1, u_2, \mathbf{x}_1, \mathbf{x}_2, d_1, d_2\right) \tag{4.1a}$$

$$\text{s.t.} \quad \mathbf{f}_1\left(u_1, u_2, \mathbf{x}_1, d_1\right) = \mathbf{0}, \tag{4.1b}$$

$$\mathbf{f}_2\left(u_1, u_2, \mathbf{x}_2, d_2\right) = \mathbf{0}, \tag{4.1c}$$

$$\mathbf{g}\left(\mathbf{u}, \mathbf{x}, \mathbf{d}\right) \leq \mathbf{0} \tag{4.1d}$$

where $\mathbf{x}_i \in \mathbb{R}^{n_{\mathbf{x}_i}}$, $d_i \in \mathbb{R}^{n_{d_i}}$, and $u_i \in \mathbb{R}^{n_{u_i}}$ are the state, disturbance, and decision variables (also referred to as manipulated variables, degree of freedom, or inputs) for subsystem i, respectively. Here, the label of $n_{\mathbf{x}_i}$, n_{d_i}, and n_{u_i} signify the respective number of states, disturbances and decision variables within subsystem i. The dynamic behaviour of each subsystem is captured by $\mathbf{f}_i$, while operational constraints on states, inputs and disturbances are imposed through the coupling constraints denoted as $\mathbf{g}$.

The cost function is defined as follows:

$$J\left(u_1, u_2, \mathbf{x}_1, \mathbf{x}_2, d_1, d_2\right) = J_1\left(u_1, u_2, \mathbf{x}_1, d_1\right) + J_2\left(u_1, u_2, \mathbf{x}_2, d_2\right) \tag{4.2}$$

where $J_i : \mathbb{R}^{n_{u_i}+n_{u_j}} \rightarrow \mathbb{R}$ is a function that denotes the local objective of subsystem i. Here, n_{u_j} is the number of decision variables from other subsystems that can influence the local objective of subsystem i. This composition illustrates that the cost function is not additively separable, as the local objectives are interdependent. This interdependence is also evident in the steady-state equality constraints of subsystem 1 (Equation 4.1b), where these constraints are influenced by input 2, u_2.

Further, the linearized coupling constraint, denoted as $\mathbf{g}$, can be expressed as follows:

$$\mathbf{g} = \mathbf{A}\mathbf{u} - \bar{\mathbf{y}} \tag{4.3}$$

Here, $\mathbf{A} = \begin{bmatrix}\mathbf{A}_1 & \mathbf{A}_2\end{bmatrix}$, where $\mathbf{A}_i \in \mathbb{R}^{n_{\mathbf{g}} \times n_{u_i}}$, represents a coupling matrix for different subsystems. Further, $\bar{\mathbf{y}} \in \mathbb{R}^{n_{\mathbf{g}}}$ is the coupling constraints, and $n_{\mathbf{g}}$ is the number of such constraints.

Each subsystem, denoted as i, may also have local constraints, which are expected to be independently managed by their respective local organizations or units. These local constraints are not explicitly shown in the problem formulation (4.1).

The goal of this work is to identify optimal solutions for problem (4.1), enabling system-wide steady-state optimization through the use of simple feedback controllers, all achieved in a distributed manner with limited information sharing.

Problem Statements

Let $\mathbf{u} = \{u_1, u_2\}$. The Lagrangian of problem (4.1), denoted as $\mathcal{L}(\mathbf{u}, \lambda)$, is:

$$\mathcal{L}(\mathbf{u}, \lambda) = J_1\left(u_1, u_2, \mathbf{x}_1, d_1\right) + J_2\left(u_1, u_2, \mathbf{x}_2, d_2\right) + \lambda^\top \mathbf{g}\left(\mathbf{u}, \mathbf{x}, \mathbf{d}\right) \tag{4.4}$$

Here, $\lambda \in \mathbb{R}^{n_{\mathbf{g}}}$ represents the Lagrange multiplier associated with the constraints $\mathbf{g}$.

Subsequently, the gradient of the Lagrangian of problem (4.1) w.r.t the the input $\mathbf{u}$ is given as:

$$\nabla_{\mathbf{u}}\mathcal{L}(\mathbf{u},\lambda) = \begin{bmatrix}\nabla_{u_1}\mathcal{L}(\mathbf{u},\lambda)\\ \nabla_{u_2}\mathcal{L}(\mathbf{u},\lambda)\end{bmatrix} = \begin{bmatrix}\nabla_{u_1}\mathbf{J}_1 + \nabla_{u_1}\mathbf{J}_2\\ \nabla_{u_2}\mathbf{J}_1 + \nabla_{u_2}\mathbf{J}_2\end{bmatrix} + \begin{bmatrix}\nabla^{\top}_{u_1}\mathbf{g}\\ \nabla^{\top}_{u_2}\mathbf{g}\end{bmatrix}\lambda \tag{4.5}$$

where $\nabla^{\top}_{u_i}\mathbf{g} = \mathbf{A}_i$. The terms highlighted in red indicate the challenges of decomposing an interactive system through dual or Lagrange decomposition method.

Therefore, the first challenge we are going to address in this Chapter is "*How to transform an interactive system to a non-interactive one, making it amenable to system decomposition using dual or Lagrange decomposition methods?*". Further, the subsequent challenge is "*Given a decomposed non-interactive systems, how to enable system-wide optimization through the use of simple feedback controllers?*".

4.3 Proposed Approach

4.3.1 Transforming Interactive to Non-Interactive System

In this section, our aim is to transform an interactive system into a non-interactive one by *identifying* of *locally* predicted solution of other subsystem. For example, the most obvious predicted solution of subsystem 2 within subsystem 1 is by introducing variable $\hat{u}_2$ within subsystem 1, and $\hat{u}_1$ as the predicted solution of subsystem 1 within subsystem 2. This concept allows us to transform and *reformulate* problem (4.1) as follows:

$$\min_{u_1,u_2,\hat{u}_1,\hat{u}_2} \quad J_1\left(u_1,\hat{u}_2,\mathbf{x}_1,d_1\right) + J_2\left(\hat{u}_1,u_2,\mathbf{x}_2,d_2\right) \tag{4.6a}$$

$$\text{s.t.} \quad \mathbf{g}\left(\mathbf{u},\mathbf{x},\mathbf{d}\right) \leq \mathbf{0}, \tag{4.6b}$$

$$g_{s,1} : u_1 - \hat{u}_1 = 0, \tag{4.6c}$$

$$g_{s,2} : u_2 - \hat{u}_2 = 0 \tag{4.6d}$$

Note that the dynamic behaviours of the subsystems are not explicitly shown in the problem formulation (4.6).

Consequently, the Lagrangian of the transformed problem (4.6) is given as:

$$\mathcal{L}(\mathbf{u},\lambda) = J_1\left(u_1,\hat{u}_2,\mathbf{x}_1,d_1\right) + J_2\left(\hat{u}_1,u_2,\mathbf{x}_2,d_2\right) + \begin{bmatrix}\lambda\\ \lambda_{s,1}\\ \lambda_{s,2}\end{bmatrix}^{\top}\begin{bmatrix}\mathbf{g}\\ g_{s,1}\\ g_{s,2}\end{bmatrix} \tag{4.7}$$

Defining $\mathbf{u} = \{u_1,\hat{u}_2,u_2,\hat{u}_1\}$, the gradient of the Lagrangian of problem (4.6) w.r.t the the input $\mathbf{u}$ is given by:

$$\nabla_{\mathbf{u}}\mathcal{L}(\mathbf{u},\lambda) = \begin{bmatrix}\nabla_{u_1}\mathcal{L}_1(\mathbf{u},\lambda)\\ \nabla_{\hat{u}_2}\mathcal{L}_1(\mathbf{u},\lambda)\\ \nabla_{u_2}\mathcal{L}_2(\mathbf{u},\lambda)\\ \nabla_{\hat{u}_1}\mathcal{L}_2(\mathbf{u},\lambda)\end{bmatrix} = \begin{bmatrix}\nabla_{u_1}\mathbf{J}_1\\ \nabla_{\hat{u}_2}\mathbf{J}_1\\ \nabla_{u_2}\mathbf{J}_2\\ \nabla_{\hat{u}_1}\mathbf{J}_2\end{bmatrix} + \begin{bmatrix}\nabla^{\top}_{u_1}\mathbf{g}\lambda\\ \nabla^{\top}_{\hat{u}_2}g_{s,2}\lambda_{s,2}\\ \nabla^{\top}_{u_2}\mathbf{g}\lambda\\ \nabla^{\top}_{\hat{u}_1}g_{s,1}\lambda_{s,1}\end{bmatrix} \tag{4.8}$$

This formulation (equation (4.8)) clearly separates problem (4.7) into two subsystems: subsystem 1 represented by the first and second lines, which include the stationary condition to predict $\hat{u}_2$, and subsystem 2 represented by the third and fourth lines, which

include the stationary condition to predict $\hat{u}_1$. This reformulation enables the use of dual or Lagrangian decomposition techniques to solve the original problem (4.1).

4.3.2 Master Controllers as Constraint and Consensus Controllers

Starting from an initial guess λ_0, as a constraint controller, the master controller takes on the role by updating the associated shadow price through iterative steps:

$$\mathbf{g} \leftrightarrow \tilde{\lambda}; \qquad \lambda = \texttt{max}\left(0, \tilde{\lambda}\right); \tag{4.9}$$

Note that, according to the KKT conditions, $\lambda \geq 0$ must hold for inequality (coupling) constraints in problem (4.6). This requirement is ensured by using a `max` operator.

Concurrently, acting as consensus controllers, and starting from an initial guess $\lambda_{s,10}$, $\lambda_{s,20}$, the master controllers iteratively update the associated shadow prices as follows:

$$g_{s,1} \leftrightarrow \lambda_{s,1}; \tag{4.10}$$

$$g_{s,2} \leftrightarrow \lambda_{s,2}; \tag{4.11}$$

The master controllers, responsible for achieving *consensus* between predicted and actual values, essentially handle equality constraints. Consequently, the associated Lagrange multipliers are non-zero, specifically $\lambda_{s,1} \neq 0$ and $\lambda_{s,2} \neq 0$.

4.3.3 Dual-based Distributed Feedback-optimizing System

Given a *non-interactive version* of the steady-state optimization problem (4.1), we aim to solve this problem using simple feedback controllers in a distributed manner. Specifically, we seek to find a self-optimizing controlled variable for each subproblem as a function of shadow prices such that the master controllers can be used to coordinate the local feedback controllers, thus achieving system-wide optimal operation. In this thesis, this framework is known as the dual-based *distributed Feedback-optimizing System (dFoS)*, as depicted in Figure 4.2. In this figure, the grey box represents the physical system, the white box represents the computation block, and v_i denotes real-time measurements in subsystem i.

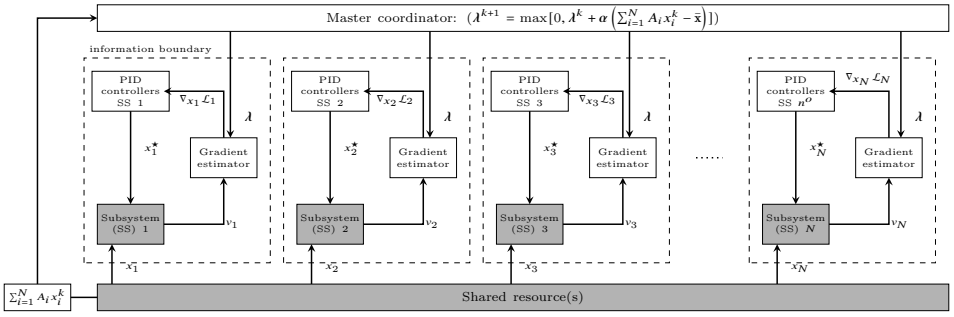


Figure 4.2: Dual-based distributed feedback-optimizing system for optimal resource allocation in a *non-interactive system* using simple feedback controllers. Adopted from [41].

From the perspective of Lagrangian decomposition, it becomes evident that, for the stationary condition of subproblem i (as seen in the first and third lines of Equation (4.8)), we

require $\nabla_{u_i}\mathcal{L}_i(u_i,\lambda)=0$. Based on this concept, the self-optimizing controlled variable of subsystem i can be expressed as a function of the Lagrange multiplier λ:

$$c_i(\boldsymbol{\lambda})=\nabla_{u_i}\mathcal{L}_i(u_i,\lambda) \tag{4.12}$$

This controlled variable needs to be regulated to a constant set-point of $c_i^{sp}=0$ to satisfy the necessary condition of optimality (NCO).

4.3.4 Feedback-optimizing Control for Solution Predictions

In this work, we also aim to *predict* solution from other subsystems using simple feedback controller in a distributed manner. Similarly, we aim to determine a self-optimizing controlled variable for each subproblem as a the shadow prices associated with consensus controllers, as described in the coupling constraints (4.6c)-(4.6d).

Within the framework of Lagrangian decomposition, achieving the stationary condition for the predictor in subproblem i (as evident in the second and fourth lines of Eq.(4.8)) necessitates $\nabla_{\hat{u}_j}\mathcal{L}_i(\hat{u}_j,\lambda_{s,j})=0$. Based on this idea, the self-optimizing controlled variable of the predictor in subsystem i can be expressed as:

$$c_{ij}(\lambda_{s,j})=\nabla_{\hat{u}_j}\mathcal{L}_i(\hat{u}_j,\lambda_{s,j}) \tag{4.13}$$

which must be controlled to a constant set-point of $c_{ij}^{sp}=0$ to satisfy the necessary condition of optimality (NCO).

The local estimation of the self-optimizing controlled variable (4.13) is feasible when no information or measurements are required from other subsystems. For instance, this applies when both the cost and constraint functions are linear w.r.t. the predicted solution. To broaden the scope of application, where the predicted solution is not limited by the degrees of freedom of other subsystems ($\hat{u}_j$), please consider the following theory:

Theorem 4.1: Intermediate predicted solution identification method

Let $\hat{u}_j$ represent the predicted solution for another subsystem, and consider the local stationary condition of subsystem i, denoted as $\nabla_{\hat{u}_j}\mathcal{L}_i(\hat{u}_j,\lambda s,j)=0$. If there exists an accessible intermediate variable y_{itm} that depends on $\hat{u}_j$, and the Jacobian of y_{itm} w.r.t to the predicted solutions has full rank, we can choose this intermediate variable, y_{itm}, as the predicted solution variable. In such a case:

$$c_{ij}(\lambda_{s,j})=\nabla_{y_{itm}}\mathcal{L}_i(y_{itm},\lambda_{s,j}) \tag{4.14}$$

Controlling c_{ij} to 0 leads to optimal operation with zero loss.

Proof : Intermediate predicted solution identification method

Consider the partial derivative $\frac{\partial y_{itm}}{\partial \hat{u}_j}$, which represents the sensitivity of the intermediate variable y_{itm} with respect to the predicted solution of another subsystem, $\hat{u}_j$. Notably, this information is not locally accessible.

Let $\frac{\partial \mathcal{L}_i}{\partial y_{itm}}$ denote the partial derivative of the Lagrange function $\mathcal{L}_i$ with respect to the intermediate variable y_{itm}. Importantly, this information is locally accessible.

Now, let's examine the chain rule for derivatives, which relates these expressions.

$$\frac{\partial \mathcal{L}_i}{\partial \hat{u}_j} = \frac{\partial \mathcal{L}_i}{\partial y_{itm}} \frac{\partial y_{itm}}{\partial \hat{u}_j}$$

It becomes clear that by controlling $\frac{\partial \mathcal{L}_i}{\partial y_{itm}}$ to zero, we achieve optimal operation. This control action is equivalent to controlling $\frac{\partial \mathcal{L}_i}{\partial \hat{u}_j}$ to zero. □

4.3.5 Local Estimators

To evaluate (4.12)-(4.13), each subsystem i must locally estimate its steady-state cost and constraint gradient using either model-based or model-free techniques, as discussed in [35] and [74]. This process is depicted in Figure (4.2), aided by the transformations detailed in Section 4.3.1 and Equation (4.8), crucial for constructing a dual-based *dFoS* for an interactive system.

4.3.6 Summary of Proposed Framework

Equation (4.12) shows that by driving $\nabla_{u_i} \mathcal{L}_i(u_i, \lambda)$ to 0 within each subsystem i by employing simple feedback controllers like PID controllers, and by iterative updates of the Lagrangian multipliers (with assumption that the iterations converge), we can avoid solving the optimization problem online.

In this framework, the master controller has limited knowledge of the individual subsystems, influencing only the Lagrangian multipliers, commonly known as shadow prices. Consequently, subsystems are relieved from the necessity of sharing intricate details such as models, measurements, local constraints, and objective functions. The only essential information to be shared pertains to constrained variables, including those employed for solution predictions.

The proposed framework eliminates the need for online numerical optimization, which means that the sampling rate is no longer limited by computation time. This flexibility allows master controllers and various subproblems to operate at different sampling rates, enhancing adaptability to system requirements. Furthermore, the framework can be implemented across subsystems with different time scales, accommodating a wide range of system configurations. Additionally, depending on the choice of the gradient estimation scheme used, the proposed scheme also avoids the steady-state wait-time issue.

In traditional distributed optimization framework, the master coordinator problem and the subproblems require several iterations to converge to the optimal solution at each time step. Meanwhile, in *dFoS*, we use real-time measurements (which are not the solution to a optimization problem), to update the Lagrange multiplier $\boldsymbol{\lambda}$ in the master controller. This process resembles real-time iteration between the master controller and the subproblems, driving the entire plant toward the correct direction for achieving a steady-state optimal.

Under assumption of *perfect* control, *dFoS* can also be shown that as $t \to \infty$, the different subproblems converges to a KKT point of problem 4.1.

4.4 Numerical Examples

In this section, we present two illustrative examples. The first highlights a static system, emphasizing the importance of solution predictors and consensus controllers in *dFoS*

(proposed framework) without intermediate variables. Conversely, the second example focuses on a dynamic system, emphasizing solution predictors and consensus controllers with intermediate variables.

4.4.1 Motivating Example

In this section, we evaluate the propose framework against the standard dual-based *dFoS* [41], and also apply primal-dual *FoC*, which does not involve system decomposition.

Problem description

Consider the following steady-state optimization problem with decision variables u_i and parameters/disturbances p_i associated with subsystems $i = 1, 2$:

$$\min_{u_1,\, u_2} \quad J = f\left(u_1, p_1, u_2, p_2\right) \tag{4.15a}$$

$$\text{s.t.} \quad g : u_1 + u_2 - 0.4 + p_3 \leq 0 \tag{4.15b}$$

where p_3 is the coupling parameter/disturbance. The overall objective function combines the local objectives of the two subsystems ($J = J_1 + J_2$), where local objectives:

$$J_1 = f_1\left(u_1, u_2, p_1\right) = 2u_1^2 - (1.5 + p_1)u_2$$

$$J_2 = f_2\left(u_1, u_2, p_2\right) = 3u_2^2 - (1.0 + p_2)u_1$$

Additionally, the state variables z_i for $i = 1, 2$ are determined by:

$$z_1 = (1.5 + p_1)u_2, \qquad z_2 = (1.0 + p_2)u_1$$

System analysis

Note that z_1 is influenced by subsystem 2's decision variable (u_2), similar to z_2 and u_1, rendering the cost function non-additively separable. When evaluating the augmented linearized gain matrix $\mathbf{G_{up}}$ for $\mathbf{u}$ and $\mathbf{p}$ to $\nabla_{\mathbf{u}}\mathcal{L}$, we find:

$$\mathbf{G_{up}} = \begin{bmatrix} 4 & 0 & 0 & -1 \\ 0 & -1 & 6 & 0 \end{bmatrix}$$

According to Definition 2.4, assuming constant parameters categorizes this system as decoupled or decomposed. However, with time-varying parameters, it becomes interactive; $\nabla_{u_1}\mathcal{L}$ is influenced by subsystem 2's parameters, and vice versa (as noted in Remark 2.8). Thus, the interactive components are the parameters or disturbances.

As a result, the overall system is interactive and non-separable. Consequently, attempting system decomposition becomes infeasible, since the objective function of each subsystem relies on other subsystems. Furthermore, highlighting the significance of the solution predictor, when ignoring interactive components, subsystem solutions become disturbance-independent.

Identification of predicted solution variables and problem reformulation

To construct a decomposable non-interactive system, we introduce predicted solution variables of other subsystems in a local subsystem. For instance, in subsystem 1, we introduce $\hat{u}_2$ as a local predicted solution variable of subsystem 2. This introduction reconfigure Problem (4.15) into the following form:

$$\min_{u_1,\,u_2} \quad f_1\left(u_1, p_1, \hat{u}_2\right) + f_2\left(u_2, p_2, \hat{u}_1\right) \tag{4.16a}$$

$$\text{s.t.} \quad g : u_1 + u_2 - 0.4 + p_3 \leq 0, \tag{4.16b}$$

$$g_1 : u_1 - \hat{u}_1 = 0, \tag{4.16c}$$

$$g_2 : u_2 - \hat{u}_2 = 0 \tag{4.16d}$$

Defining $\mathbf{u} = \{u_1, \hat{u}_2, u_2, \hat{u}_1\}$, the gradient of the Lagrangian of problem (4.16) w.r.t the the input $\mathbf{u}$ is given as:

$$\nabla_{\mathbf{u}}\mathcal{L}(\mathbf{u},\lambda) = \begin{bmatrix} \nabla_{u_1}\mathcal{L}_1(\mathbf{u},\lambda) \\ \nabla_{\hat{u}_2}\mathcal{L}_1(\mathbf{u},\lambda) \\ \nabla_{u_2}\mathcal{L}_2(\mathbf{u},\lambda) \\ \nabla_{\hat{u}_1}\mathcal{L}_2(\mathbf{u},\lambda) \end{bmatrix} = \begin{bmatrix} 4u_1 \\ -(1.5+p_1) \\ 6u_2 \\ -(1.0+p_2) \end{bmatrix} + \begin{bmatrix} \lambda_g + \lambda_{g_1} \\ -\lambda_{g_2} \\ \lambda_g + \lambda_{g_2} \\ -\lambda_{g_1} \end{bmatrix} \tag{4.17}$$

where the associated Lagrange multipliers $(\lambda_g, \lambda_{g_1}, \lambda_{g_2})$ will be updated using master controllers, and distributed to all subsystems. As a result, predicted solution variables eliminate dependencies of subsystem 1 on u_2 and subsystem 2 on u_1.

Control structure in the proposed framework

Figure 4.3 shows the control structure of proposed framework. The solid black line PID

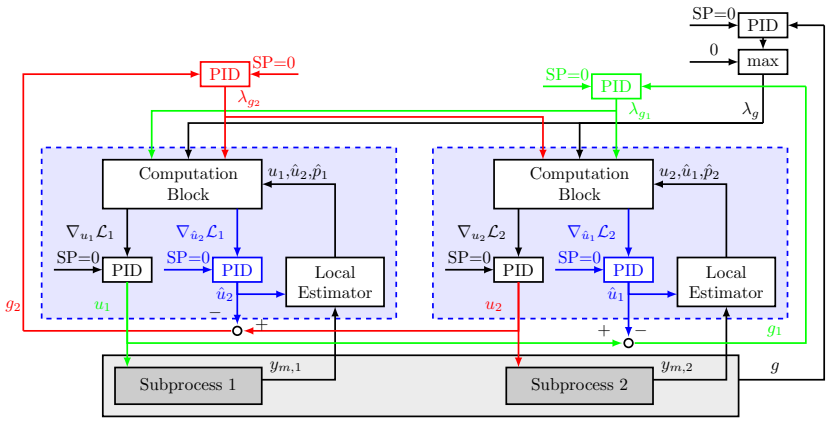


Figure 4.3: Proposed control structure for motivating example

controller with a `max` selector (on the top right) is a master controller from [41], acting as a constraint controller with automatic active constraint switching. Inside the light blue box, PID controllers with solid black lines represent the original local self-optimizing controllers from [40]. Solid blue line PID controllers serve as solution predictors (providing

information for local estimator), adjusted by master controllers for consensus with actual values. The solid red and green line PID controllers are master controllers (as consensus controllers) aligning predicted and actual values. All controllers use SIMC tuning [52]. Controller details are in Table 4.1-4.2.

Table 4.1: Parameters used in the local controllers shown in Figure 4.3

Tuning Par.	Local Grad. Controller 1	Solution Predictor 1	Local Grad. Controller 2	Solution Predictor 2
K_I	0.1250	0.0833	0.0833	0.1250

Table 4.2: Parameters used in the master controllers shown in Figure 4.3

Tuning Par.	Constraint Controller	Consensus Controller 1	Consensus Controller 2
K_P	-0.7937	-1.4297	-0.0420
K_I	-0.1984	-0.0275	-0.0008
K_{AW}	0.4000	–	–

Parameter or disturbance estimation technique

We employ a model-based estimation technique within each local estimator. Mathematically, this approach is represented as:

$$\hat{p}_1 = \frac{\left(2u_1^2 - J_1\right)}{\hat{u}_2} - 1.5 \tag{4.18a}$$

$$\hat{p}_2 = \frac{\left(3u_2^2 - J_2\right)}{\hat{u}_1} - 1.0 \tag{4.18b}$$

Note that $\hat{u}_2$ is locally provided by the solution predictors feedback locally within subsystem 1, while $\hat{u}_1$ is similarly provided within subsystem 2. Meanwhile, we estimate $\hat{p}3 = g - utot + 0.4$, where $u_{tot} = u_1 + u_2$. This estimation is performed immediately using measured constrained variables.

Simulation results

Figure 4.4 demonstrates that our proposed framework, with solution predictors and consensus controllers, achieves steady-state optimal profits as the primal-dual FoC can achieve. In contrast, the original dual-based *dFoS* (without solution predictors), not only falls short in profit but also loses its sensitivity to disturbances residing within interactive components. This compelling illustration emphasises the necessity of incorporating solution predictors and consensus controllers.

Equation (4.18) shows that without predictors, predicted variables, i.e., $\hat{u}_1$ and $\hat{u}_2$ will not be updated. In the original dual-based *dFoS* framework, this leads to inaccurate local disturbance estimation. However, Figure 4.5 demonstrates that solution predictors and consensus controllers allow each subsystem to autonomously estimate parameters/disturbances locally.

Figure 4.6 demonstrates the proposed framework's ability to control and switch active constraint regions. Notably, the proposed framework, along with the original dual-based

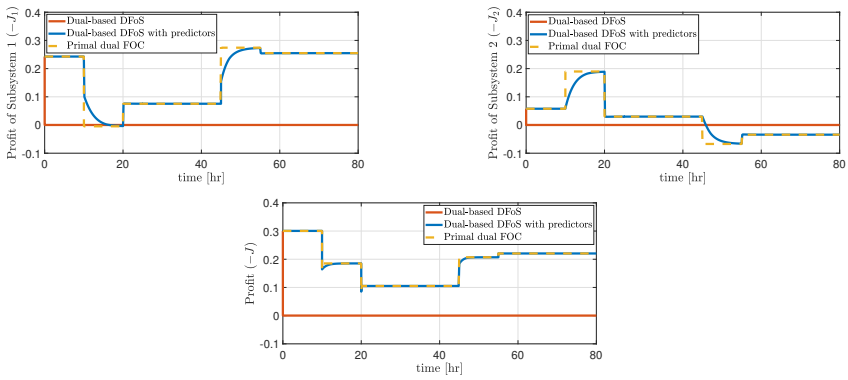


Figure 4.4: Simulation results: Profits

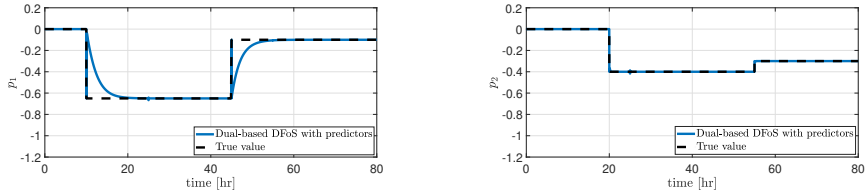


Figure 4.5: Simulation results: Disturbance/Parameter Estimation

dFoS, operates on a slower time scale for constraint control compared to primal-dual FoC. This becomes apparent from $t = 10$ to $t = 15$ hours, where shadow price updates are slow, resulting in significant dynamic violations in the proposed dual-based *dFoS* framework.

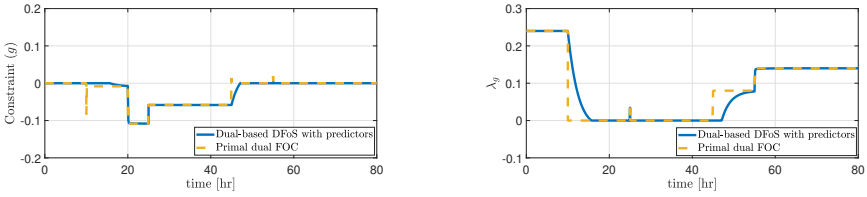
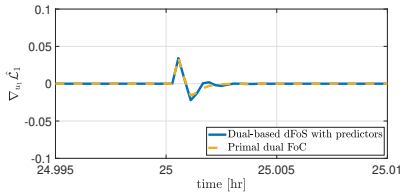


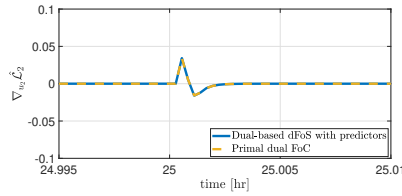
Figure 4.6: Simulation results: CV - MV of coupling constraint controllers

Figures 4.7(a)-4.7(b) display stationary states achieved with local gradient controllers, while Figures 4.7(c)-4.7(d) depict similar outcomes for the *auxiliary* stationary states with solution predictors. Figure 4.8 verifies consensus between predicted and actual solutions achieved using consensus controllers.

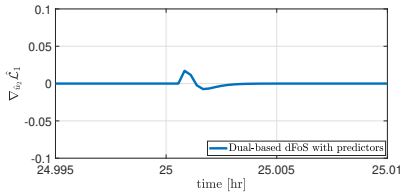
In summary, simulation results from Figures 4.4 to 4.8 indicate the promise of the proposed framework in decomposing interactive systems and achieving system-wide steady-state optimal performance. In the next example, we demonstrate its effectiveness in a more realistic case study.



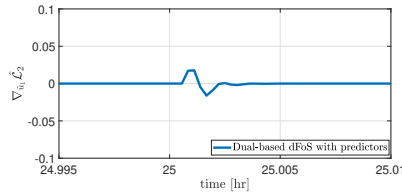
(a) Controlled by grad. cont. in Subsystem 1



(b) Controlled by grad. cont. in Subsystem 2

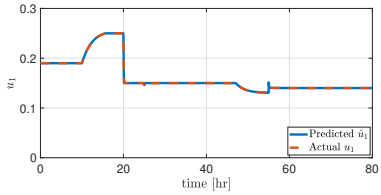


(c) Controlled by sol. pred. in Subsystem 1



(d) Controlled by sol. pred. in Subsystem 2

Figure 4.7: Simulation results: Stationary conditions



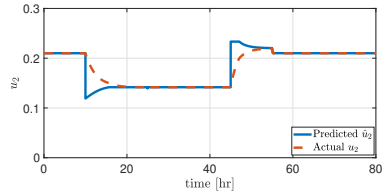


Figure 4.8: Simulation results: Consensus of predicted and actual solutions

4.4.2 Interactive Gas Lift Optimization

Problem description

In this section, we apply the proposed framework on a subsea gas-lifted oil production well network, including the riser (see Fig. 4.9) with $N = 2$ subsystems (clusters), operated by two independent production units. Unit i operates well i within cluster i for $i = 1, 2$.

We evaluate the proposed framework by comparing it with the original dual-based *dFoS* [41] and a primal-dual *FoC*, serving as a benchmark. The overall objective is to optimize revenue from oil production and minimize gas lift compression costs. The two production units, operating different wells, share a common topside process facility as illustrated in Figure 4.9. However, this facility has capacity limitations, restricting the total gas that can be exported. This limitation on total gas production is a key coupling constraint among the subsystems, resulting in the following optimization problem:

$$\min_{w_{gl,1},\ldots,w_{gl,N}} \quad -\$_o \sum_{i=1}^{N} w_{oe,i} + \$_{gl} \sum_{i=1}^{N} w_{gl,i} \tag{4.19a}$$

$$\text{s.t.} \quad w_{ge}^{tot} - \bar{w}_{ge} \leq 0 \tag{4.19b}$$

where $\$_o$ is the oil price, and $\$_{gl}$ is the cost of gas compression. Further, the total gas export, $w_{ge}^{tot} = \sum_{i=1}^{N} w_{ge,i}$.

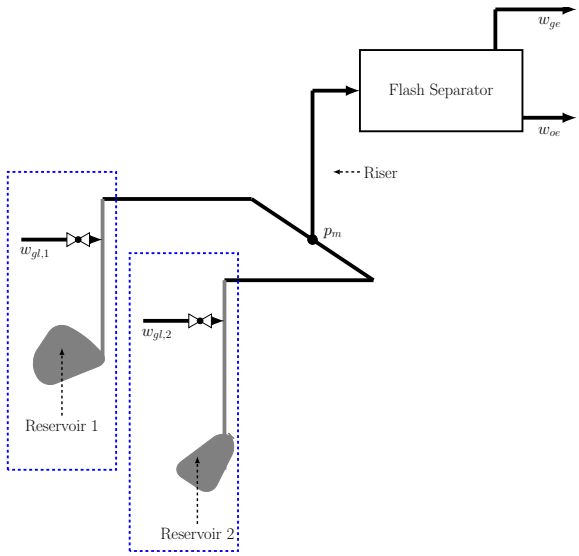


Figure 4.9: A simplified process diagram of a subsea gas-lifted oil production well network.

In this context, we consider the gas-lift injection rate, denoted as $w_{gl,i}$, as the decision variable. Additionally, we have exported oil and gas production rates, $w_{oe,i}$ and $w_{ge,i}$, respectively, which depend on the gas-lift injection. The local objective function is represented as:

$$J_i = -\$_o w_{oe,i} + \$_{gl} w_{gl,i}$$

Furthermore, the gas-to-oil ratio (GOR), a reservoir property, serves as a time-varying disturbance affecting different wells (feed disturbance). In this simulation study, we assume variable GOR values for different wells. A high GOR indicates that a well contains lighter fluid, requiring a smaller gas-lift injection rate compared to wells with lower GOR values to produce the same amount of oil.

Primal Dual Feedback-optimizing Control

In a centralized manner (undecomposed system), units typically share information across the entire system, making primal-dual *FoC* a suitable choice. For this case study, we employ the same control structure used in Chapter 2, as shown in Figure 2.17. Note that, when employing model-based parameter and gradient estimation, a good complete model of the entire system is required.

Dual-based Distributed Feedback-optimizing System

Efficient production operations typically entail sharing only effective information, avoiding the inefficiency of sharing all data. For instance, Gas-Oil Ratio (GOR) typically need not be distributed beyond its relevant production operation unit. This concept is visually depicted in Figure 4.9 by limiting information sharing within the dashed blue line. Con-

sequently, the system necessitates decomposition into two independent production units.

Furthermore, an offshore riser is required for hydrocarbon transport from subsea to surface facilities. Therefore, we may also need one additional local subsystem with limited information sharing. Yet, from an optimization standpoint, we assume no degrees of freedom for optimal decisions within the riser.

To achieve distributed optimal operation, we control the proposed local self-optimizing variable (4.12). However, decomposing the two wells into separate production units challenges the assumption of an additively separable cost function. This challenge arises because the production from one well can impact the other due to factors like commingle pressure, such as manifold pressure. To address this interaction and enable the independent operation of production units, we control the manifold pressure to a constant value. The setpoint can be obtained from the results of a prior and slower time-scale optimization.

Given a constant pressure manifold, in this example, (4.12) can be represented as,

$$\nabla_{w_{gl,i}} \mathcal{L}_i(w_{gl,i}, \lambda_{ge}) = \nabla_{w_{gl,i}} \hat{J}_i + \left(\nabla_{w_{gl,i}} \hat{w}_{ge,i}\right)^\top \lambda_{ge} \tag{4.20}$$

where the Lagrange multiplier (also known as dual variable or shadow price) associated with constraint (4.19b) is labelled as λ_{ge}.

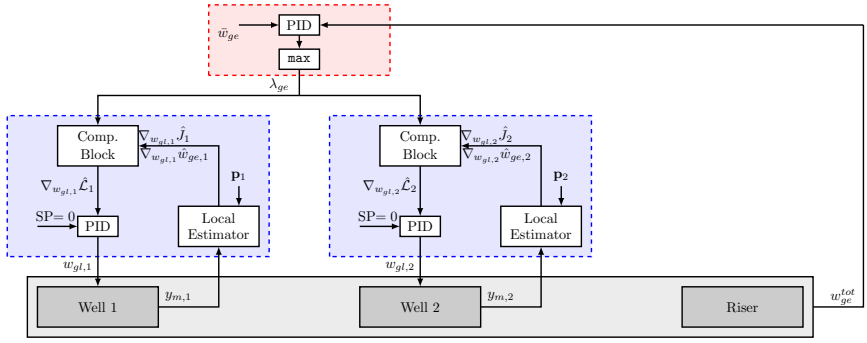


Figure 4.10: Dual-based *dFoS* control structure for interactive gas lift optimization problem.

In Figure 4.10, we illustrate the control structure of the original dual-based *dFoS* implemented in the interactive gas lift optimization problem. Notably, the local feedback-optimizing algorithm within the light blue box exclusively relies on local measurements from well 1 (managed by production unit 1) and dual variables from the constraint controller (inside the light red box). No information exchange occurs with the other unit.

Furthermore, in the context of model-based parameter estimation, the local estimator requires a set of constant local parameters, denoted as $\mathbf{p}_1$, forming the foundation for first-principle modeling techniques. Within this set, one crucial element is the manifold pressure, represented as p_m.

Unlike primal-dual *FoC*, our dual-based *dFoS* does not require a riser model for model-based parameter and gradient estimation. Only the local well model is required.

Dual-based Distributed Feedback-optimizing System with Solution Predictors

As mentioned, the key strategy of the proposed framework involves identifying predicted solution variables to equalize with the actual ones. An obvious candidate is the

optimal gas-lift flow rate of the other production unit. However, this candidate proves impractical for prediction due to its dependence on measurements from the other unit.

To address this challenge, we utilize Theorem 4.1 and introduce an intermediate predicted solution variable. In this context, the manifold pressure stands out as the ideal candidate, denoted as $\hat{p}_{m,i}$ for unit i. Instead of regulating the manifold pressure to a constant value, our local optimization problem treats this variable as a "decision variable." This approach enables the incorporation of a feedback-based solution predictor to predict the optimal value of the actual manifold pressure, locally managed by the riser system. Consequently, the local steady-state optimization problem takes the form:

$$\min_{w_{gl,i},\,\hat{p}_{m,i}} \quad -\$_o w_{oe,i} + \$_{gl} w_{gl,i} \tag{4.21a}$$

$$\text{s.t.} \quad w_{ge,i} - \bar{h}_i \leq 0, \tag{4.21b}$$

$$\hat{p}_{m,i} - p_m = 0 \tag{4.21c}$$

where the new decision variables are $\mathbf{u}_i = \begin{bmatrix} w_{gl,i} & \hat{p}_{m,i} \end{bmatrix}^\top$, and the actual manifold pressure, p_m, is locally managed by riser system operation unit. Further, $\bar{h}_i$ is any constant value but it will be eliminated after the first derivative.

Define $\mathbf{g}_i = \begin{bmatrix} g_i(1) & g_i(2) \end{bmatrix}^\top = \begin{bmatrix} w_{ge,i} - \bar{h}_i & \hat{p}_{m,i} - p_m \end{bmatrix}^\top$, and $g = w_{ge}^{tot} - \bar{w}_{ge}$. Let λ_g and λ_{g_i} be the associated Lagrange multipliers of constraint g and $g_i(2)$, respectively. Figure 4.11 illustrates the implementation of a dual-based distributed feedback-optimizing system with solution predictors in the interactive gas lift optimization problem. Solid blue line and PIDs in blue are the additional control structure that incorporate solution predictors (inside light blue boxes) and consensus controllers (inside light red boxes).

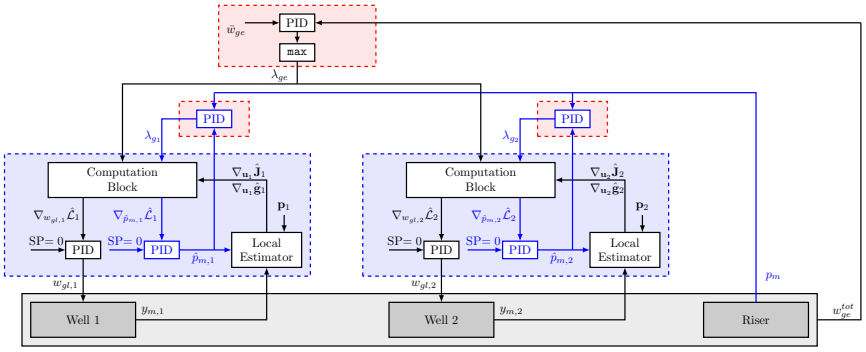


Figure 4.11: Proposed framework: Dual-based distributed feedback-optimizing system with solution predictors for interactive gas lift optimization problem.

Figure 4.11 shows that the predicted solution variables, $\hat{p}_{m,i}$, are locally updated and informed to local estimator. These updates facilitate more accurate disturbance and parameter estimation within each unit. Notably, this estimation process operates independently, requiring no actual information exchange with other operation units. Instead, it relies on the local Lagrange multipliers from the consensus controllers. These multipliers enable control actions to align the predicted variables with the actual values managed by the riser operation unit.

Most importantly, while predicted solution variables are formulated as decision variables, there is no requirement for additional manipulated variables, such as pressure controller, in the actual process. The values of these variables are only transmitted to the local estimator.

In this example, the self-optimizing control variable linked to the solution predictor (4.13) variable, $\hat{p}_{m,i}$, can be defined as,

$$\nabla_{\hat{p}_{m,i}} \mathcal{L}_i(w_{gl,i}, \hat{p}_{m,i}, \lambda_{ge}, \lambda_{g_i}) = \nabla_{\hat{p}_{m,i}} \hat{J}_i + \lambda_{g_i} \tag{4.22}$$

The solution predictor variable, $\hat{p}_{m,i}$, is continuously updated in real-time to drive the self-optimizing control variable (4.22) to 0.

Given the real-time update manifold pressure, the self-optimizing control variables (4.12) can be represented as in equations (4.20). This approach operates without needing a riser model for the model-based parameter and gradient estimation. It only requires the local well model. This is possible because there is no degree of freedom in the riser system. In a broader system context, the manifold pressure variable acts as artificial boundary (that need to be updated properly), connecting two sequential small optimization problems (serial interconnection). Nevertheless, we do not explore this concept further to maintain focus on this chapter's topic.

Parameter Estimation

Local estimators, whether incorporated in dual-based *dFoS* with or without solution predictors and consensus controllers, or an integrated estimator within primal-dual *FoC*, share a common two-step process. In the first step, disturbance estimation takes place, facilitated by a comprehensive measurement set. This stage employs the extended Kalman filter [26, 75], resulting in the estimation of disturbance, differential, and algebraic states.

Given both the decision variables and the estimated disturbance, as well as the differential states provided in the first step, the second step involves the estimation of the steady-state gradients of cost and constraint functions. In this chapter, we adopt a model-based gradient estimation framework proposed by [55]. Note that this framework is not restricted to this specific gradient estimation approach, as alternative model-based or model-free gradient estimation schemes [35] can be employed instead.

PID Controllers

In this study, we use PID controllers that are tuned using the SIMC tuning rules recommended by [52]. These controllers are designed with a sampling time of 1 second. For controller tuning and parameter values utilized in this simulation, see Table 4.3-4.4.

Table 4.3: Parameters used in the local controllers shown in Figure 4.11

Tuning Par.	Local Grad. Controller 1	Solution Predictor 1	Local Grad. Controller 2	Solution Predictor 2
K_P	11.9688	–	11.1624	–
K_I	0.0199	1.2332	0.0186	1.2110

Table 4.4: Parameters used in the master controllers shown in Figure 4.11

Tuning Par.	Constraint Controller	Consensus Controller 1	Consensus Controller 2
K_P	-0.0240	$-8.8457e-04$	$-9.0683e-04$
K_I	$-1.2413e-04$	$-7.6919e-07$	$-7.8040e-07$
K_{AW}	19.3346	–	–

Simulation Setup

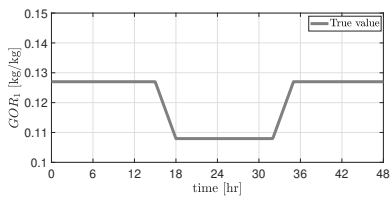
The overall plant is modelled as a DAE (Differential Algebraic Equations). The model equations can be found in Appendix A, and the model parameters used in this case study are shown in Table 4.5. The plant simulator is developed using the `CasADi ver. 3.5.1` toolbox ([53]) in `MATLAB R2019b` and is simulated using the `IDAS` integrator.

Table 4.5: List of wells and riser parameters and their corresponding values in the simulation.

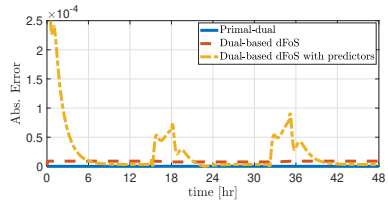
Par. [units]	Well 1	Well 2	Par. [units]	Riser
L_w [m]	1500	1500	L_r [m]	400
H_w [m]	1000	1000	H_r [m]	300
D_w [m]	0.121	0.121	D_r [m]	0.121
L_{bh} [m]	500	500	-	-
H_{bh} [m]	500	500	-	-
D_{bh} [m]	0.121	0.121	-	-
L_a [m]	1500	1500	-	-
H_a [m]	1000	1000	-	-
D_a [m]	0.189	0.189	-	-
ρ_o [kg/m^3]	800	800	ρ_{or} [kg/m^3]	800
C_{iv} [m^2]	1E-4	1E-4	C_{pr} [m^2]	1E-2
C_{pc} [m^2]	2E-3	2E-3	-	-
p_r [bar]	150	155	p_s [bar]	20
PI [kg/bar.s]	3.5	3.5	-	-
T_a [^{o}C]	28	28	T_r [^{o}C]	30
T_w [^{o}C]	32	32	-	-

Simulation Results

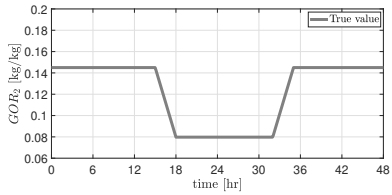
The Gas-Oil Ratio (GOR) exhibits variations, as shown in Figure 4.12(a) and 4.12(c). These fluctuations introduce disturbances affecting optimal gas-lift allocation. To assess GOR estimation accuracy, we examine the absolute error obtained by primal-dual *FoC*, original dual-based *dFoS*, and the proposed framework, as displayed in Figure 4.12(b) and 4.12(d). These figures highlight the effectiveness of the proposed framework in minimizing steady-state estimation errors. In contrast, the absence of predictors and consensus controllers results in persistent steady-state errors.



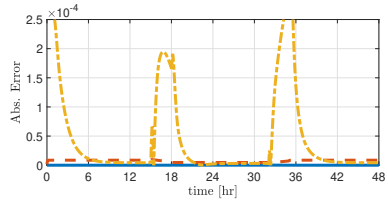
(a) True GOR in Subsystem 1



(b) Abs. error of estimated GOR in Subsystem 1



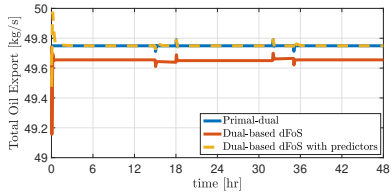
(c) True GOR in Subsystem 2



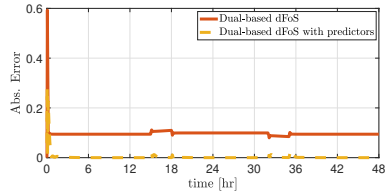
(d) Abs. error of estimated GOR in Subsystem 2

Figure 4.12: Simulation results: True and Estimated Disturbances (GOR), along with the absolute error between them (in the right-hand plots)

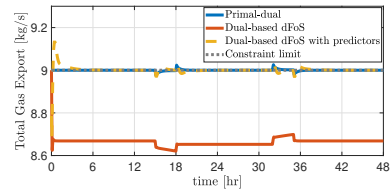
Figure 4.13 presents simulation results, comparing primal-dual *FoC*, original dual-based *dFoS*, and the proposed framework. The absolute error between the steady-state optimum and both the original dual-based *dFoS* and the proposed framework is shown in Figure 4.13(b) and 4.13(d) for total exported oil and gas. These figures illustrate the proposed framework's successful convergence to the steady-state optimum, whereas the original dual-based *dFoS* cannot eliminate steady-state errors.



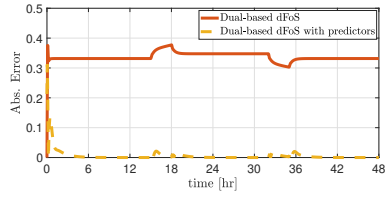
(a) Total oil exported



(b) Abs. error of total oil exported



(c) Total gas exported



(d) Abs. error of total gas exported

Figure 4.13: Simulation results: the performance of original *dFoS* and proposed framework compared to the steady-state optimum provided by primal-dual *FoC*, along with the absolute error between them (in the right-hand plots)

The original dual-based *dFoS* fails to eliminate steady-state errors due to the absence of a feature that drives predicted variables toward optimal solutions, essential for eliminating interaction effects. Figure 4.14 illustrates the success of the proposed framework in locally driving predicted variables (manifold pressures) to optimal solutions. Conversely, without predictors and consensus controllers, these variables remain unupdated.

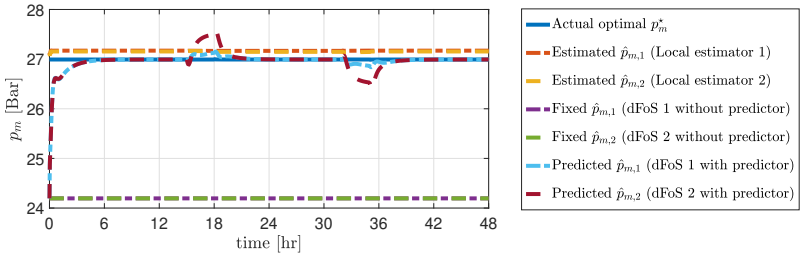
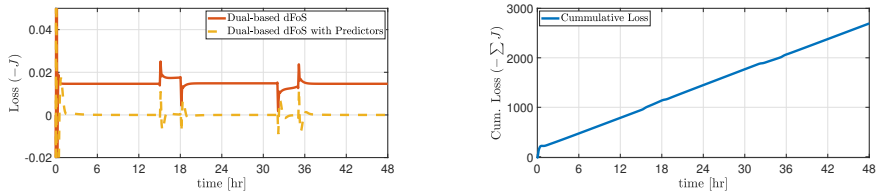


Figure 4.14: Simulation results: Manifold pressure

Note that one might consider treating the unknown manifold pressure as a disturbance and estimating it through a suitable parameter estimation technique. However, as shown in Figure 4.14, this approach has drawbacks:

- It does not resolve the challenge of equalizing a *key* controlled variable to enable system decomposition, leading to differing estimations across subsystems in steady-state.
- It leads to less accurate disturbance estimation, relying solely on local measurements.

To the end, the proposed framework eliminates steady-state losses (Figure 4.15(a)), while the original *dFoS* experiences cumulative losses that rise over time (Figure 4.15(b)).



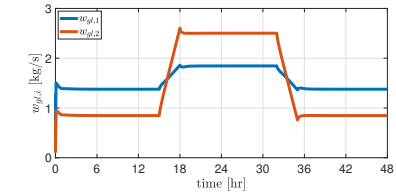
(a) Profit loss of both *dFoS* without and with predictors and consensus controllers.

(b) Cumulative loss of *dFoS* without predictors and consensus controllers.

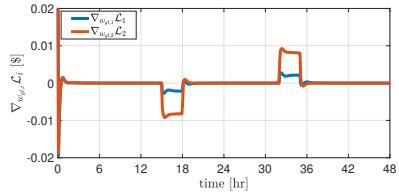
Figure 4.15: Simulation results: the profit's performance of *dFoS* and proposed framework (with predictors and consensus controllers).

Figure 4.16 presents further simulation results for the proposed framework. Figure 4.16(c) highlights how, from $t = 18 - 32$ h, constraint (4.19b) relaxes, leading to a decrease in Lagrange multipliers λ_{ge}. After $t = 35$ h, the gas processing capacity constraint tightens. Additionally, Figure 4.16(d) shows that varying GOR influences the optimal dual

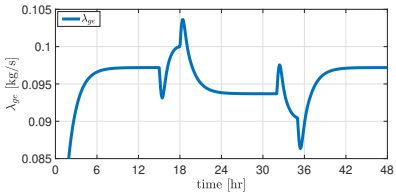
variables of the equality constraints associated to the manifold pressure as the predicted solution variable.



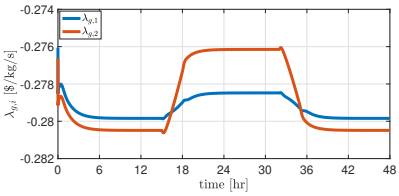
(a) Gas lift allocation among wells



(b) Stationary conditions among wells



(c) Shadow price of gas export constraint



(d) Shadow price of consensus constraints

Figure 4.16: More simulation results from the proposed framework

Discussions

During the transition, note that as GOR changes, the associated Lagrange multipliers is converging in real-time, as seen in Figure 4.16(c). This may lead to temporary dynamic violations of the total gas production rate, but these constraints are satisfied at steady-state. Severe dynamic constraint violations can occur during the coupling constraint switch from unconstrained to constrained scenarios, as shown in [41], primarily due to slow time-scale constraint control.

Regarding manifold pressure local estimation, one might consider having the riser operation unit directly transmit actual manifold pressure to each well subsystem. While this strategy is feasible and effective, it contradicts the principle of limited information sharing and deviates from the general system decomposition concept.

4.5 Chapter Summary

As introduced by [41], this chapter starts with a dual-based *dFoS* to decompose non-interactive systems. This framework employs simple feedback controllers to control (4.12) to a constant setpoint of zero. As (4.12) is a function of Lagrange multipliers, we showed that by utilizing a master controller (with a `max` selector) for Lagrange multiplier updates, it enables system-wide optimal operation.

Additionally, to address previous chapter concerns, we incorporate a dual-based *dFoS* with solution predictors, featuring consensus controllers. This approach allows system decomposition for interactive systems. The key idea involves predicting solution variables and equalizing them, through consensus controllers, with measured variables, thereby enabling the decomposition of large-scale interactive systems. For serial interconnection systems, it enables real-time artificial boundary updates.

To the end, this proposed framework allows different time scale separations among subsystems, mitigates numerical robustness issues, minimizes information sharing between subsystems, and enables system-wide optimal operation. It is compatible with both model-based and model-free gradient estimation schemes, making it widely applicable.

Assessment Table:

Let S-IX be the proposed framework of this chapter, the parameter we assess and compare are described in section 2.5. Table 4.6 summarizes the assessment , where the

Table 4.6: Summary of Chapter 4

	S-IV	S-VI	S-IX
(a)	Yes	Yes	Yes
(b)	Yes	Yes	Yes
(c)	Yes	Yes	Yes
(d)	No[1]	No[1]	No[1]
(e)	No	Yes	Yes
(f)	No (Medium)	Yes	Yes
(g)	Yes	No[2]	Yes
(h)	No	Yes	Yes
(i)	No	No	No
(j)	Yes	No	No

"Yes": the approach has satisfied the success parameter
"Yes": the approach has satisfied the success parameter in this Chapter
"No": the approach does not satisfied the success parameter
"No": does not satisfied the success parameter, and will be addressed in this thesis.
[1]: is addressed in Chapter 3
[2]: is addressed in Chapter 4
"No (High)": does not satisfied the success parameter (f) at all (shown in Chapter 2).
"No (Medium)": has partly satisfied the success parameter (f) better than "No (High)".

approaches are:

- S-IV: Dual-based hybrid RTO
- S-VI: Primal dual FOC
- S-IX: Dual-based Distributed Feedback-Optimizing System

and the parameters we assess and compare are:

- (a): Is steady-state optimal achieved?
- (b): Is constraint controlled *transparently* (see Definition 2.1)?
- (c): Is flexible for active constraint switching?
- (d): Is constraint controlled *directly* (on fast time scale)?
- (e): Is applicable for less than twice differentiable Lagrange function?
- (f): Is numerical solver avoidable (computation time)?
- (g): Is it recommended for complex and large system?
- (h): Is input filter (or additional setpoint controller) avoidable?
- (i): Is parameters and states dynamic estimator avoidable?
- (j): Is gradient estimator avoidable?

Future works: Transforming interactive problems into non-interactive ones is crucial for enabling system decomposition and implementing the dual-based *dFoS* framework. Experimental validation is discussed in Chapter 5. Beyond practical benefits, this transformation enhances clarity for subsequent analyses, such as integrating override controllers, ensuring proper pairing of override controller, designing multi-input override controllers, and completing the algorithm with auxiliary constraint controllers. These additional studies are addressed in Chapters 6 and 7, respectively.

Chapter 5

Experimental Validation of Dual-based DFoS Framework in a Gas-lifted Oil Well Rig

The goal of this chapter is to experimentally validate the approach described in Chapter 4. This chapter is based on the work in [65]

5.1 Introduction

Many process systems feature multiple parallel units or subsystems coupled by stream of energy or material flows, as highlighted in various studies [76, 77, 78, 66]. For instance, in large industrial operations, a common power plant might supply steam to various subprocesses [76, 78, 79]. Similarly, in oil production networks, multiple reservoirs' oil wells may utilize the same topside processing facility [80, 81, 82, 41]. In such scenarios, the preference is often to break down and optimize subprocesses locally, as distributed decision-making tools prove more practical and manageable in comparison to large-scale centralized optimization.

To meet this requirement, we proposed the dual-based distributed feedback-optimizing system (dual-based *dFoS*) framework, with its advantages detailed in [41] and Chapter 4. This approach, designed for linear coupling constraints, has been analytically demonstrated to converge to the stationary point of the overall optimization problem under reasonable assumptions [40]. This approach has been applied in a simulation study to a large-scale subsea production system with both linear and nonlinear coupling constraints, and enabling active constraints switching in [41], where it was shown that the dual-based *dFoS* framework was able to drive the system to its overall optimal operation without the need to repeatedly solve numerical optimization problems online.

Remark 5.1: Benefit 1 of dual-based dFoS framework

It is important to emphasize that dual-based *dFoS* framework mainly consists of master controllers and local gradient controllers, which can be implemented using simple tools such as PID controllers. Therefore, this framework can achieve optimal performance in the steady-state without any numerical optimization solver.

Remark 5.2: Benefit 2 of dual-based dFoS framework

It is interesting to note that the estimation of the local steady-state cost and constraint gradients in each subsystem needs only the models and the real-time process measurements of the local subsystem. This is a useful property that enables implementation of dual-based *dFoS* framework in systems where minimum data exchange are important as motivated in [66, 40] for example.

Building on the previous work [40, 41], Chapter 4, we now experimentally validate this approach on a lab-scale gas-lifted oil well rig consisting of three wells with the lift gas being the shared resource that couples the three parallel wells together [83]. To this end, the goal of this chapter is to experimentally validate the dual-based dual-based *dFoS* framework ([41], Chapter 4) and benchmark its performance with a numerical optimization based RTO.

5.2 Experimental Setup

In subsea production systems, wells are located on the seabed to extract the hydrocarbons trapped in the underground reservoir. The produced oil and gas is transported along the seabed in pipelines to the processing facility where the riser pipeline takes it from the seabed to the surface. If the reservoir pressure is low, either naturally or due to depletion, artificial lift methods may be needed to overcome the pressure losses and bring the hydrocarbons to the surface. Gas-lift is a commonly used artificial lift method, in which compressed gases are injected into the well tubing to reduce the fluid mixture density and, consequently, the hydrostatic pressure losses.

However, injecting too much gas increases the frictional pressure drop in the well tubing, which has a counteracting effect [60]. Therefore, each well has a local optimum corresponding to the gas lift injection rate. The total available lift gas is often a limited resource that must be optimally allocated among the wells to maximize the production from the overall production network. For more practical information of production optimization in oil and gas industry, the reader is referred to [14, 84] and the references therein.

5.2.1 Experimental rig as a subsea production system

To mimic the subsea gas-lifted oil production system, we use a lab-scale experimental rig that uses water and air as working fluids instead of oil and gas for simplification. The choice of working fluids does not influence the gas lift phenomenon, which can still be observed in the lab rig. Thus, the rig is suited for studying production optimization methods, where the gas lift effect is the phenomenon of interest.

Figure 5.1 shows a simplified flowsheet of the system indicating three sections: a reser-

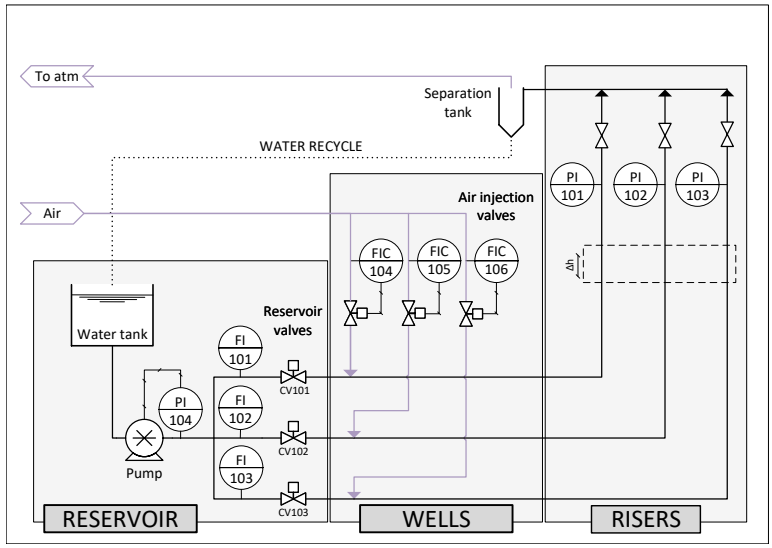


Figure 5.1: Experiment schematic. Adapted from [83].

voir, well, and riser section. The system measurements, denoted by $\mathbf{y_p}$, include well top pressures (PI101, PI102, and PI103), pump outlet pressure (PI104), liquid flowrates (FI101, FI102, and FI103), and gas flowrates (FI104, FI105, and FI106). Three PI controllers are used to control the gas flowrates, denoted by $\mathbf{u} = \begin{bmatrix} Q_{gl,1} & Q_{gl,2} & Q_{gl,3} \end{bmatrix}^\top$, to the calculated setpoints represented by $\mathbf{u}^{sp} = \begin{bmatrix} Q^{sp}_{gl,1} & Q^{sp}_{gl,2} & Q^{sp}_{gl,3} \end{bmatrix}^\top$. The reservoir valve openings (CV101, CV102, and CV103) represent the system disturbances. These openings are adjusted during the experiments to simulate different reservoir behaviors while keeping the pump outlet pressure constant through a PI controller.

The reservoir section contains a stainless steel tank, a centrifugal pump, and the three control valves (CV101, CV102, and CV103). These valves are used to represent disturbances from reservoir (for example, to emulate pressure oscillations, or reservoir depletion). With this setup, the reservoir produces only liquid with outflow rates ranging from 2 L/min to 15 L/min. Flow meters (FI101, FI102, and FI103) are located before the reservoir valves to measure the outflow rates. The pump's outlet pressure (PI104) is kept constant at 0.3 barg in this experiment using a PI controller that adjusts the pump rotation.

The wells consist of three parallel flexible hoses with 2 cm inner diameters and length of 1.5 m. Approximately 10 cm after the reservoir valves, pressurized air at approximately 1 barg is injected by three air flow controllers (FIC104, FIC105, and FIC106) within the range of 1 sL/min to 5 sL/min.

The risers are three vertical pipelines, orthogonal to the well section, with 2 cm inner diameters and 2.2 m high. We measure the pressures on top of the risers (PI101, PI102, and PI103). After the sensors, three manual valves are kept open during the experiments. The air is vented out to the atmosphere, and the liquid is recirculated to the reservoir water tank. More detailed description of the test setup can be found in [83]. The experimental lab rig we use in this work are shown in Figure 5.1.

5.2.2 Optimization problem setup

The objective of the optimization problem in this experimental setup is to maximize the network liquid flow rate (i.e. the summation of the liquid production of the three wells) given a limited amount of gas-lift injection. This objective can be expressed as an economic objectives as follows,

$$\begin{aligned} J(\mathbf{u}, \mathbf{p}) &:= \sum_{i=1}^{3} f_i(u_i, p_i) \\ &= -20Q_{l,1}(u_1, p_1) - 25Q_{l,2}(u_2, p_2) - 30Q_{l,3}(u_3, p_3) \end{aligned} \tag{5.1}$$

where $Q_{l,1}$, $Q_{l,2}$, and $Q_{l,3}$ are the produced liquid flowrates of wells 1, 2, and 3, respectively. For illustration, we assume that the wells have different hydrocarbon prices as shown above. The input vector is given by

$$\mathbf{u} = \begin{bmatrix} Q_{gl,1} & Q_{gl,2} & Q_{gl,3} \end{bmatrix}^\top$$

where $Q_{gl,1}$, $Q_{gl,2}$, and $Q_{gl,3}$ are the injected gas flowrates of wells 1, 2, and 3, respectively. Figure 5.2 and 5.3.

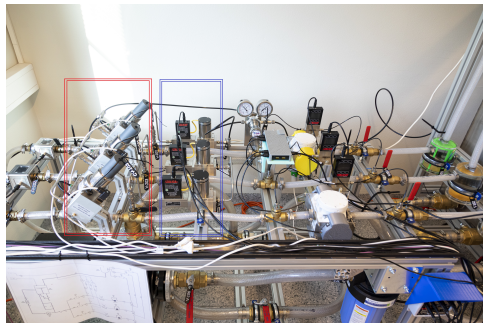


Figure 5.2: The equipment inside the red box are Reservoir valves, and inside the blue box are the Injection valves.

In the context of optimization, these flowrates are the decision variables. Meanwhile, for the plant, these flowrates are the setpoints that need to be tracked. As shown in Figure 5.1, the experimental lab rig has flow indicator and controllers (FICs) 104, 105, and 106 to regulate the air injection flowrates to their setpoints. To differentiate to the actual one, we denote these decision variables of the optimization problem as $\mathbf{u}^{sp} = \begin{bmatrix} Q^{sp}_{gl,1} & Q^{sp}_{gl,2} & Q^{sp}_{gl,3} \end{bmatrix}^\top$. Furthermore, three elements of $\mathbf{p}$, which are the reservoir valve openings CV101, CV102, and CV103, are time-varying. This implies that the cost is also a function of $\mathbf{p}$.

Considering the valve opening of FICs as decision variables presents practical challenges due to valve non-linearity and hysteresis. Instead, using injected gas flow rates as decision variables enables system decomposition and serves as a strategy to transform an interactive system into a non-interactive one through vertical decomposition. This approach is specifically tailored to parallel systems. For a more general technique, refer to Chapter 4.

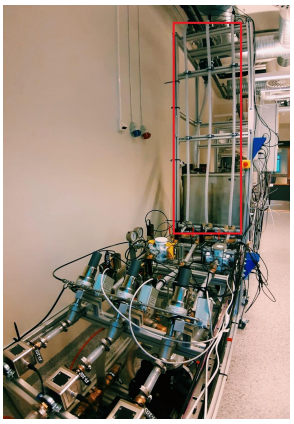


Figure 5.3: The equipment inside the red box are the risers.

The total gas availability, which is a shared (input) constraint, can also be expressed as follows:

$$\begin{aligned} g\left(\mathbf{u},\mathbf{p}\right) &:= \sum_{i=1}^{3} g_i(u_i,p_i) - g^{max} \\ &= Q_{gl,1} + Q_{gl,2} + Q_{gl,3} - Q_{gl}^{max} \end{aligned} \tag{5.2}$$

where we directly measure the constraint, and we use FICs to drive $Q_{gl,i}$ to $Q_{gl,i}^{sp}$. As these controllers are typically fast, we can approximate $Q_{gl,i} \approx Q_{gl,i}^{sp}$. To the end, the steady-state optimization problem of this rig can be expressed as follows:

$$\min_{\mathbf{u}^{sp}} \quad J\left(\mathbf{u}^{sp},\mathbf{p}\right) \tag{5.3a}$$

$$\text{s.t.} \quad g\left(\mathbf{u}^{sp},\mathbf{p}\right) = 0 \tag{5.3b}$$

Remark 5.3: Equality Constraint

Note that the total gas availability in general is an inequality constraint. However, in this experimental setup, we found that the gas lift constraint is always active, (which is also common in many gas-lifted oil fields). Hence we can equivalently consider the coupling constraint to be an equality constraint, and then we do not need the max selector for $\boldsymbol{\lambda}$ update in the constraint control.

5.2.3 Dual-based Distributed Feedback-optimizing System Setup

Control structure design

We now implement the control structure of the dual-based distributed feedback-optimizing system (*dFoS*) for our experimental setup. Given that we have three wells in the experimental setup, we decompose the problem into three subsystems. For each subsystem, we

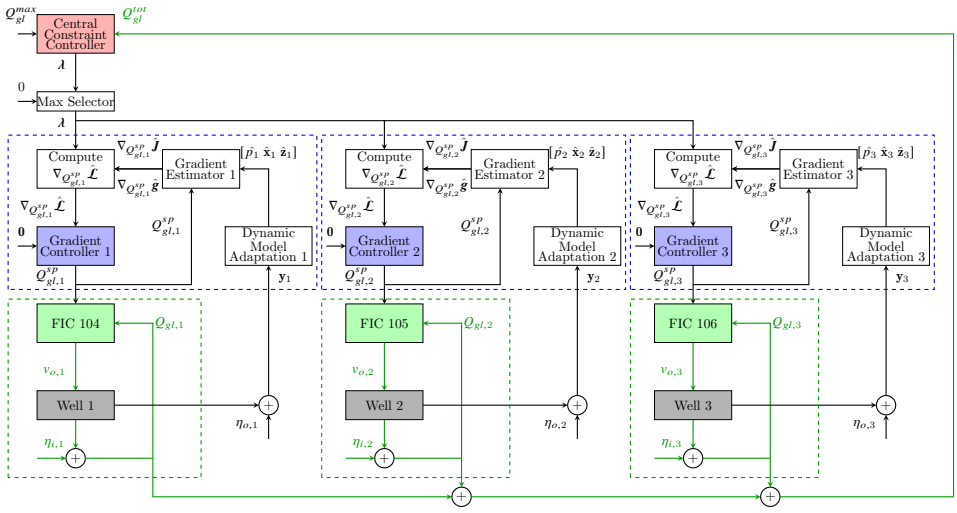


Figure 5.4: The control structures of the dual-based *dFoS* framework implemented in the experimental lab rig.

use a local gradient estimator that estimates the gradient of the local cost $\nabla Q_{l,i}$ and the constraint $\nabla Q_{gl,i}$ (cf. Remark 1). Each susbsystem has a local PI controllers that controls $c_i(\lambda)$ (see Equation 4.12) to 0 (gradient controller). The output from the gradient controller, denoted as $Q^{sp}_{gl,i}$, is used as the setpoint for the flow controller FIC$_i$ to adjust the air injection valve opening.

The control structure implemented in the lab-rig is illustrated in Figure 5.4. The diagram has three subproblems because the economic objective (5.1) is additively separable. The plant defined in Figure 5.1 consists of well 1, well 2, well 3, FIC 104, FIC 105 and FIC 106, which are inside dashed green lines in this diagram. The actual gas-lift flowrates $(Q_{gl,i})$ are states in the plant. The decision variables for dual-based *dFoS* scheme are the gas-lift flowrates setpoints $\left(Q^{sp}_{gl,i}\right)$. Using the flow controllers, the real manipulated variables for the well rig are the valve openings, labelled by $v_{o,i}$. The well rig experiences both input and output measurement noises, which are labeled as $\eta_{i,i}$ and $\eta_{o,i}$, respectively. Using the local measurement set $(\mathbf{y}_i)$ and the local dynamic model adaptation, we estimate differential states $(\hat{\mathbf{x}}_i)$, algebraic states $(\hat{\mathbf{z}}_i)$, and parameters/disturbances, $(\hat{p}_i)$. In this work, the "Dynamic Model Adaptation" is an extended Kalman filter.

Regarding the master or central constraint controller, we can directly measure the constraint g value for updating the Lagrange multiplier. However, we have to estimate both cost and constraint gradient for the gradient controllers (i.e primal controllers). We use forward sensitivity analysis to estimate these values since a reliable dynamic model of the system is available. To compute the local sensitivities, we also need to estimate the current states of the system (both differential and algebraic). Here, an extended Kalman filter (EKF) is used in each subsystem to estimate the states using only the local measurements. However, any suitable dynamic estimator can be applied, as long as it provides accurate estimates of the states and effectively filters the measurement noise properly.

Controller tuning

Once the control structure is defined, we need to tune the controllers. In the experimental rig, the fastest possible sampling rate of the data acquisition software is 1 seconds. In theory, we could execute both the gradient and the master/central constraint controllers at the same rate. However, depending on their tuning, they might compete against each other, which might drive the system to instability. Therefore, we need a time scale separation between these controllers. Given the cascaded structure, and need for proper timescale separation for smooth operation, we now provide in depth discussion on how the controllers were tuned.

The idea of dual decomposition is that the subproblems (represented by the dotted boxes in Fig. 5.4) are solved for each update (iteration) of the central coordinator problem (represented by the master/central constraint controller). However, the subproblems controlling $c_i(\lambda)$ to a setpoint of c_i^{sp} are also solved by iteration, so in practice the subproblems will not reach full convergence to their setpoint of c_i^{sp} within each central coordinator problem update. Fortunately, it is possible to estimate the approach to convergence when solving the equations using feedback controllers, as in this paper.

The rate of convergence to the setpoint in each subproblem is given by the closed-loop time constant τ_{c_i} of the corresponding control loop. More specifically, for a linear first-order system, the approach to convergence (or steady state) is $(1 - e^{-t/\tau_{c_i}})$ where τ_{c_i} is the closed-loop time constant of the i^{th} control loop, and t is the convergence time of the central coordinator problem. Thus, the approach to convergence increases from 63% to 95.0% to 99.3 % as t/τ_{c_i} increases from 1 to 3 to 5. This implies that at 5 time constants the approach is 99.3%, and convergence (or steady state) has for practical purposes been reached. This may be regarded as the basis for the rule of thumb of requiring a time scale separation between control layers of at least 5 [46]. If the time scale separation gets too small, typically 3 or less, the layers will start interacting and we may experience undesired oscillatory behavior or even instability [85]. A larger value (larger than 5) allows for robustness to process gain variations which will affect the closed-loop time constants of the control loops. However, with a too large value, the overall convergence (including the central constraint controller) will be slow, so for practical purposes, a value for the time scale separation of 5 to 10 is often recommended.

The limiting case of infinite time scale separation corresponds to $\epsilon = \tau_{c_i}/t \to 0$, which is the singular perturbation condition in the mathematical literature. Note that a time scale separation between 5 and 10, corresponds to ϵ between 0.2 and 0.1.

Remark 5.4: Timescale separation

Recently, it was shown analytically in [40] that the dual-based *dFoS* framework is guaranteed to converge to the stationary point of the overall optimization problem under the assumption of perfect control of the subproblems. This assumption can be satisfied by using a timescale separation between 5 and 10. This implies sufficiently small step for subgradient updates, and it is suggested to start with the slowest timescale separation.

In summary, the constraint should be controlled in a slow timescale ($\tau_{\lambda,c}$), and the gradient in a fast timescale ($\tau_{\mathbf{u},c}$), where we typically select the ratio to be 5 to 10. In

this paper, we use integral controllers that are tuned using SIMC-rules [52]. For the master/central constraint controller the integral gain is given by

$$\boldsymbol{\alpha} = \frac{1}{K_{\lambda}\left(\tau_{\lambda,c} + \theta_{\lambda}\right)} \tag{5.4}$$

where K_{λ} and θ_{λ} are the step response and the time delay of the constraint by the dual variable (Lagrange multiplier), and $\tau_{\lambda,c}$ is the time scale that governs the evolution of $\mathbf{g}$. For the three local gradient controllers, the integral gain is given by

$$K_{I,i} = \frac{1}{K_{\mathbf{u}_i}\left(\tau_{\mathbf{u}_i,c} + \theta_{\mathbf{u}_i}\right)} \tag{5.5}$$

where $i = 1, 2, 3$ is the well index, $K_{\mathbf{u}_i}$ and $\theta_{\mathbf{u}_i}$ are the step response and the time delay of the gradient by the primal variable (Decision variables/inputs), and $\tau_{\mathbf{u}_i,c}$ is the time scale that governs the evolution of $c_i(\boldsymbol{\lambda})$.

To determine K_{λ}, θ_{λ}, $K_{\mathbf{u}_i}$, and $\theta_{\mathbf{u}_i}$, we analyze the step responses. Meanwhile, $\tau_{\mathbf{u}_i,c}$ and $\tau_{\lambda,c}$ are the tuning parameters that should carefully consider the concept of time scale separation, where $\epsilon_i = \frac{\tau_{\mathbf{u}_i,c}}{\tau_{\lambda,c}}$ should be less than 1, which implies that the outer loop has slower time scale than the inner one. We can ideally choose $\tau_{u_i,c} = 1$ since want to drive the inner loop to the steady-state as fast as possible. However, it may be too aggressive. Therefore, we adjust them (i.e., the controller's parameter tuning) based on our practical justification and observation.

Besides avoiding "too aggressive" controllers, we also consider the fact that local gradient controllers are not the lowest in the hierarchy (see Figure 5.4), and thus the timescale of the gradient controllers should be slower than the plant (that contains the FICs and the lab rig).

The type of PID Controller of FICs is designed by the manufacture, where the valve drive is calculated based on the following PID controller equation.

$$\delta v_{o,i}^{k} = I_{FIC,i}^{k-1} + I_{FIC,i}^{k} - \frac{K_{P,FIC,i}}{65.536} Q_{gl,i}^{k} \tag{5.6a}$$

$$I_{FIC,i}^{k} = \frac{K_{I,FIC,i}}{65.536}\left(Q_{gl,i}^{sp} - Q_{gl,i}^{k}\right) \tag{5.6b}$$

where $I_{FIC,i}$ is the integral value, $K_{P,FIC,i}$ is the proportional gain, $K_{I,FIC,i}$ is the integral gain, and $\delta v_{o,i}$ is the valve drive. This controller contributes in creating plant dynamic in which the time constant may vary between 5-10 seconds.

Table 5.1 shows the controller and tuning parameters that we obtain. Note that the largest ϵ_i is 0.2113, indicating that the time-scale ratio of the overall experimental oil rig system is still within the acceptable condition of the time-scale separation concept.

Controller tuning validation

Before implementing the dual-based *dFoS* framework in the experimental rig, we first validated the controller tunings in a lab rig model developed in MATLAB. The model is a high-fidelity dynamic model of the rig, that includes the lower layer controller dynamics, i.e. FICs, input and output noise. The noise was tuned according to the information obtained from the rig. This modeling structure implies that both the lower layer controller

Table 5.1: Controller and Tuning parameters

Description	Variable	Value
Experimental rig sensors sampling time	T_s	1 second
HRTO		
Execution periods	Δt_{HRTO}	10 seconds
HRTO Input filter gain	K_u	0.4
EKF parameters	see Codes in Github	
Dual-based *dFoS*		
Execution periods	Δt_{PD}	2 seconds
Central Constraint Controller step length	α	0.0117
Gradient Controller Input 1 gain	$K_{I,1}$	0.0769
Gradient Controller Input 2 gain	$K_{I,2}$	0.0444
Gradient Controller Input 3 gain	$K_{I,3}$	0.0893
FIC 104 Proportional Gain	$K_{P,FIC,1}$	8560
FIC 105 Proportional Gain	$K_{P,FIC,2}$	8560
FIC 106 Proportional Gain	$K_{P,FIC,3}$	8560
FIC 104 Integral Gain	$K_{I,FIC,1}$	100
FIC 105 Integral Gain	$K_{I,FIC,2}$	100
FIC 106 Integral Gain	$K_{I,FIC,3}$	50

dynamics and the noise are part of the plant as presented by the diagram block shown in Figure 5.4. The reader is referred to the code available on our Github page[1] for detailed parameters. In addition, [83] contains a detailed description of the model.

Remark 5.5: Simulator for Controller Tuning

Note that the simulator model of the test rig is only used to determine the controller tuning parameters, before it is implemented on the actual rig. All results that are presented later in Section 5.3 are from the actual experimental rig, and not the simulator.

Benchmark method

To benchmark the performance of the distributed feedback-based RTO approach, this paper considers the Hybrid Real-time Optimization (HRTO) [2] that solves the centralized numerical optimization problem. We choose the HRTO approach as our benchmark since this circumvents the steady-state wait time issue in traditional steady-state RTO (see [10] and [11]). Simply put, HRTO continuously estimates the parameters (and differential states in general cases) using dynamic models and transient measurements, e.g. by applying an extended Kalman filter. Then, the corresponding updated steady-state model is used for computing the solution of the economic optimization problem. In our experimental setup, we use the exact same state and parameter estimator for the HRTO and our proposed DFRTO scheme. The optimal setpoints $\mathbf{u}^{sp}$ computed by the HRTO layer are given to the flow controllers FIC. We consider HRTO with the execution period of 10

[1] https://github.com/Process-Optimization-and-Control/ProductionOptRig

[2] HRTO is the same as RTO with persistent parameter adaptation (ROPA) [11]

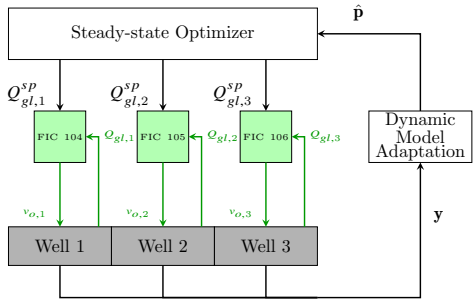


Figure 5.5: Block diagram for Hybrid RTO (HRTO) Illustration.

seconds in this experiment. Figure 5.5 illustrates this method. The grey box represents a given plant. The white boxes represent computational blocks. In this work, the "Dynamic Model Adaptation" is an Extended Kalman Filter.

To summarize, the HRTO and dual-based *dFoS* framework differ in the fact that in HRTO, a steady-state optimization problem is solved to determine the optimal setpoint u^{sp}, whereas in dual-based *dFoS* framework, the optimal setpoints u^{sp} are given by the feedback controllers. Thus, HRTO is a good benchmark for dual-based *dFoS* framework.

Steady-state Gradient Estimation

This paper uses forward sensitivity analysis to estimate the gradient. This gradient estimation has two main steps. First, we use the current plant information to update the state and parameters of the model using a dynamic adaptation scheme (here, extended Kalman filter). Next, we use the updated model to compute the gradients via the forward sensitivity analysis.

Remark 5.6: System Model

The system has a differential-algebraic model, where $\mathbf{x} \in \mathbb{R}^{n_\mathbf{x}}$ is a vector of differential states, $\mathbf{z} \in \mathbb{R}^{n_\mathbf{z}}$ is a vector of algebraic states, and $\mathbf{y} \in \mathbb{R}^{n_\mathbf{y}}$ is a vector of output (measured states). This model is necessary for state and parameter estimation.

In order to use the Kalman filter equations, we first linearize the available model. Note that the model is a differential-algebraic equation (DAE) system; however, since it is an index-1 model, it can be easily re-arranged into an ordinary differential equation (ODE). Additionally, we assume that the unknown parameters are time-varying. Their dynamics are determined by a random walk model:

$$\mathbf{p}_{k+1} = \mathbf{p}_k + \mathbf{v}_k \tag{5.7}$$

where $\mathbf{v}^k$ follows a normal distribution with mean zero and covariance V_θ. Additionally, we assume that the increments $\mathbf{v}^k$ are independent of $\mathbf{v}^{\neq k}$.

Then, we combine the system dynamics and parameter dynamics to obtain an extended model that is used for parameter estimation. Since the model was linearized, we can apply extended Kalman filter equations for estimating $\mathbf{x}^k$, $\mathbf{z}^k$, and $\mathbf{p}^k$ simultaneously. For a complete derivation of the EKF equations, please refer to [86].

To use forward sensitivity analysis, consider the original nonlinear DAE model in the following form:

$$\begin{aligned} \mathbf{x}^{k+1} &= \breve{F}(\mathbf{x}^k, \mathbf{z}^k, \mathbf{u}^k; \mathbf{p}^k) \\ 0 &= \breve{G}(\mathbf{x}^k, \mathbf{z}^k, \mathbf{u}^k; \mathbf{p}^k) \end{aligned} \tag{5.8}$$

The steady-state gradients are estimated using the stationary value of forward sensitivity equations:

$$\begin{aligned} 0 &= \frac{\partial \breve{F}^\top}{\partial \mathbf{x}} S_{SS} + \frac{\partial \breve{F}^\top}{\partial \mathbf{z}} R_{SS} + \frac{\partial \breve{F}^\top}{\partial \mathbf{u}} \\ 0 &= \frac{\partial \breve{G}^\top}{\partial \mathbf{x}} S_{SS} + \frac{\partial \breve{G}^\top}{\partial \mathbf{z}} R_{SS} + \frac{\partial \breve{G}^\top}{\partial \mathbf{u}} \end{aligned} \tag{5.9}$$

where S_{SS} and R_{SS} are the sensitivities of the differential states $\mathbf{x}$ and algebraic states $\mathbf{z}$ w.r.t. the inputs $\mathbf{u}$.

Since, in our specific case, the objective J and constraint function $\mathbf{g}$ are linear functions of the algebraic states ($J = \breve{H}_J z$, $\mathbf{g} = \breve{H}_\mathbf{g} z$), we use the chain rule to obtain $\nabla_\mathbf{u} J$ and $\nabla_\mathbf{u} \mathbf{g}$:

$$\begin{aligned} J = \breve{H}_J z \implies \nabla_\mathbf{u} J = \breve{H}_J R_{ss} \\ \mathbf{g} = \breve{H}_\mathbf{g} z \implies \nabla_\mathbf{u} \mathbf{g} = \breve{H}_\mathbf{g} R_{ss} \end{aligned} \tag{5.10}$$

5.3 Experimental Results and Discussions

Utilizing the control and tuning parameters shown in Table 5.1, we implemented the dual-based *dFoS* framework and the HRTO to serve as benchmark. Figure 5.6 shows the reservoir valve openings ($CV101$, $CV102$, $CV103$) that we consider as the disturbance in this experiment. The first disturbance occurs when the opening of $CV101$ gradually decreases from $t = 5$ to $t = 11$ minutes. We expect a decrease in the gas-lift injection in well 1, and a redirection of the gas supply to the other wells. The second disturbance occurs

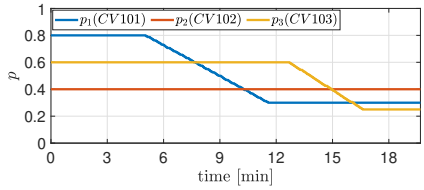


Figure 5.6: The change of reservoir valve openings (CV101, CV102, and CV103) during the experiments for representing different reservoir behaviors.

when the opening of $CV103$ also gradually decreases from $t = 13$ to $t = 16.5$ minutes. We expect that the gas supply to well 3 reduces with larger rate since the "hydrocarbon price" of this well is higher. Meanwhile the other wells will obtain more gas supply with larger rate as well. We try to avoid sudden disturbance to ensure that the controller can adjust the plant smoothly.

In the rig, we used a programming environment (LABVIEW [87]) to automate the implementation of these disturbance. Therefore, it is possible to repeat the independent experiments with the same disturbance profile.

Comparison of the optimal setpoints

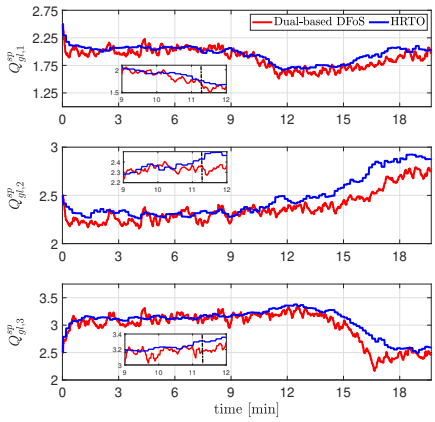


Figure 5.7: The gas-lift flow rate setpoint $\left(\mathbf{u}^{sp} = \mathbf{Q}_{gl}^{sp}\right)$ of every wells due to reservoir parameter changing (disturbance) from the Experimental Lab Rig.

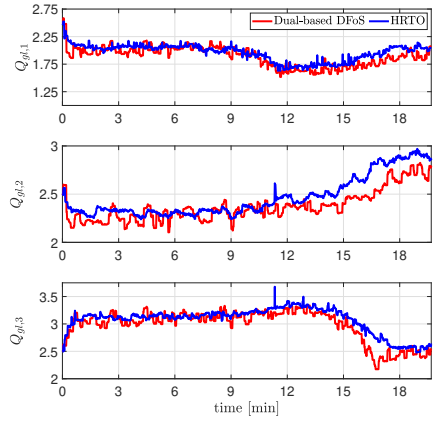


Figure 5.8: The measured gas-lift flow rate $(\mathbf{u})$ of every wells due to reservoir parameter changing (disturbance) from the Experimental Lab Rig.

We first run the experimental results comparing dual-based *dFoS* framework and HRTO, which are presented in Figure 5.7-5.9. Figure 5.8 depicts the actual gas-lift flow rate. The measured trajectories are slightly different from the calculated input setpoint shown in Figure 5.7. This difference occurs due to input measurement noise and the fact that the gas flowrates controllers (FIC 104, FIC 105, FIC 106, see Figure 5.1) need time to settle the actual gas-lift flow rate $\mathbf{u}$ to the setpoint of gas-lift flow rate $\mathbf{u}^{sp}$. Sometimes the setpoint change calculated by HRTO is quite significant such that the gas flowrates controllers res-

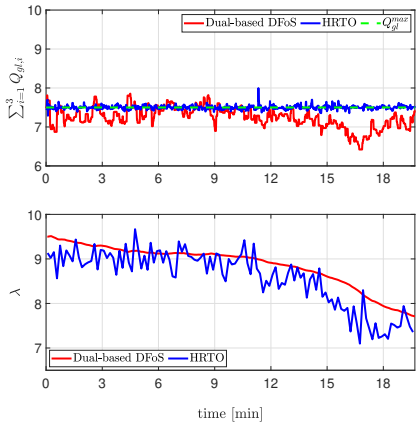


Figure 5.9: Constraint satisfaction and Lagrange multiplier evolution due to reservoir parameter changing (disturbance) from the Experimental Lab Rig.

ults in input spikes (see around $t = 11$ min in Figure 5.8). The number of these spikes is reduced because we have implemented first order input filter in the setpoints computed by HRTO, i.e.:

$$\mathbf{u}^{sp,k+1} = \mathbf{u}^{sp,k} + K_u \left(\mathbf{u}^{sp,\star,k} - \mathbf{u}^{sp,k}\right)$$

where $\mathbf{u}^{sp,\star}$ is the optimal setpoint given by HRTO solver. Meanwhile, dual-based *dFoS* framework does not have the input filter because it has gradient controller(s) as input filter(s).

Figure 5.7-5.8 show that the dual-based *dFoS* framework is slower in responding to the second disturbance (see around $t = 15$ min). This slow response is the consequence of considering time-scale separation concept explained in Section 5.2.3. Specifically, each plot in Figure 5.7 has a magnifying plot in the time window 9 to 12 minutes. These plots show that when disturbance p_1 start settling down at around 11 hrs, the calculated input setpoints $\mathbf{u}^{sp}$ seem slightly off initially.

Figure 5.9 shows the constraint satisfaction and its associated Lagrange multiplier, where the initial guess of the Lagrange multiplier is slightly off in dual-based *dFoS* framework. The Lagrange multiplier λ of the dual-based *dFoS* framework is around 9.5 at $t = 0$ min. Meanwhile, the mean value of Lagrange multiplier λ of HRTO is around 9.1 at that time. However, the dual-based *dFoS* framework slowly drives the Lagrange multiplier to converge to a better value that is closer to HRTO solver obtains. We can observe the effect of the more accurate λ estimate on the performance of the dual-based *dFoS* framework, since, around $t = 6$ minutes, the active constraint is slightly better controlled. In any case the difference here is not significant, and the variations are mainly due to measurement noise.

From around $t = 12$ min, the dual-based *dFoS* framework converges slower than HRTO following the second disturbance. This is mainly due to the tuning of the contraint controller, where it can be seen in Figure 5.8 that the λ converges slower than the HRTO. As λ converges, the dual-based *dFoS* converges to the same value as HRTO at around

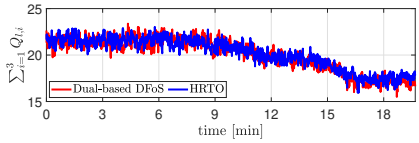


Figure 5.10: Total production comparison due to reservoir parameter changing (disturbance) from the Experimental Lab Rig

$t = 18$ min, confirming that dual-based *dFoS* is able to povide the same adymptotic performance as HRTO. This behavior is expected as we carefully design the master/central constraint controller. By considering the timescale separation concept, we avoid undesired behaviors, such as oscillatory behavior. On the other hand, the master/constraint control performance becomes relatively slow, and thus any dynamic constraint violation may last quite some time. Since we expressed the total gas capacity constraint as an equality constraint (see Equation (5.2)), the dynamic constraint violation also includes the situation when the total of gas-lift flowrates is less than Q_{gl}^{max}.

Due to various process and measurement noise, tightly controlling the hard constraints maybe a challenging task for the dual-based *dFoS* (which may need a back-off depending on the noise levels). Nevertheless, the dual-based *dFoS* still drives the system to the optimal steady-state value.

In addition, the control structure of the dual-based *dFoS* allows the possibility to have other sources of error such as additional control dynamic from master/central constraint coordinator, gradient estimation error, transmission error, and measurement error. Any error (due to improper tuning or design) from one of them leads to the additional disturbance for constraint satisfaction.

Surprisingly, although the Lagrange multiplier calculated by HRTO is sometimes slightly off and less smooth than dual-based *dFoS*, HRTO has a relatively good constraint satisfaction. This condition happens because the constraints are on the inputs. Therefore, even a high degree of plant-model mismatch does not influence the constraint satisfaction performance. In other words, the constraint model ends up with a simple summation of a 'known' input. Therefore, HRTO may have a better constraint satisfaction performance in handling an input constraint in this specific case.

To summarize, by observing Figure 5.7-5.10, we can conclude that the solutions of the two compared approaches, i.e., dual-based *dFoS* and HRTO are similar in this experiment run.

Average Values

To ensure reproducability, we then re-run the experiments once again with the same set of disturbance profile for the different approaches. Figure 5.11 compares the input setpoint rate $\Delta Q_{gl,i}^{sp} = Q_{gl,i}^{sp,k} - Q_{gl,i}^{sp,k-1}$ of the implemented approaches from two independent experiments.

Figure 5.11 also shows that the input rate setpoints of dual-based *dFoS* are, in general, comparable to HRTO. However, we can still note that for well 1 and 3, HRTO has significant outliers (-0.1 for well 1 and around 0.15 for well 3). These outliers might occur due to a numerical optimizer with an imperfect numerical condition or a bad parameter

estimation. In an extreme case, the HRTO may suffer from numerical robustness issues.

Figure 5.12 shows that the average inputs setpoint trajectory resulting from both approaches are similar over time. Based on these results, we can conclude that dual-based *dFoS* has a similar performance to HRTO in general, which also supports the same observation stated in the previous results. Figure 5.12 also shows that in the time window $t = 8$ to $t = 14$ minute, the trajectories of average input solutions of dual-based *dFoS* seem mild. However, after $t = 14$ minute, the input trajectories need to be more aggressive to achieve optimal performance. This condition seems to create challenges for dual-based *dFoS* as it may take more time to obtain the optimal performance.

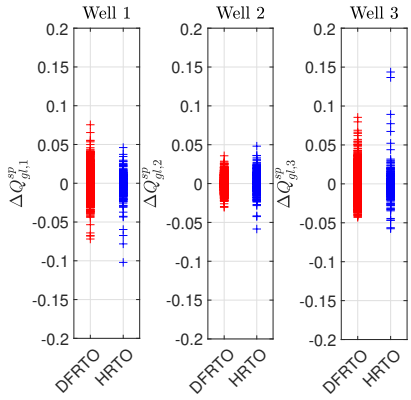


Figure 5.11: The comparison of $\Delta Q^{sp}_{gl,i} = Q^{sp,k}_{gl,i} - Q^{sp,k-1}_{gl,i}$ of each well due to reservoir parameter changing (disturbance) from the Experimental Lab Rig

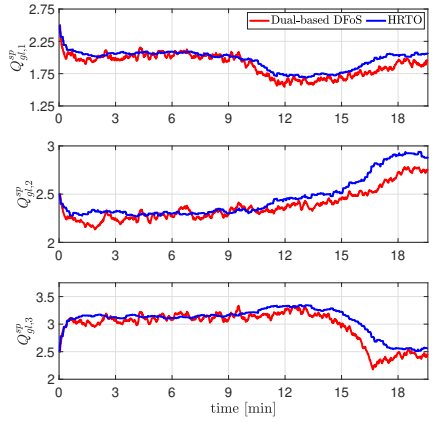


Figure 5.12: The comparison of average inputs setpoint trajectories from the Experimental Lab Rig

Comparison of the optimal cost

To analyze the optimization performance of dual-based *dFoS* and HRTO, we compare the profit (from both individual experiments) obtained by the two approaches with the naive approach, where we consider fixed inputs, i.e.,

$$\mathbf{u} = \begin{bmatrix} Q_{gl,1}^{sp} & Q_{gl,2}^{sp} & Q_{gl,3}^{sp} \end{bmatrix}^\top = \begin{bmatrix} \frac{Q_{gl}^{max}}{3} & \frac{Q_{gl}^{max}}{3} & \frac{Q_{gl}^{max}}{3} \end{bmatrix}^\top$$

The naive approach illustrates the case in which no information about the system is available. Hence, the best alternative is to divide the available gas equally among the wells. This latter approach is another benchmark to show that the optimization methods, i.e, dual-based *dFoS* and HRTO, will give more profit compared to not doing any optimization at all.

To measure the performance, we plot the difference, in percentage, between the instantaneous profit of the approach of interest (i.e. dual-based *dFoS* or HRTO) and the naive approach. The difference is calculated as

$$J_{diff} = \frac{J - J_{naive}}{J_{naive}}.100 \tag{5.11}$$

where J is the profit of the approach of interest, and J_{naive} is the profit of the naive approach. In addition, we use a 60s moving average for smoothing the profiles, because the instantaneous profit measurements are noisy.

Figure 5.13 shows that dual-based *dFoS* and HRTO are more profitable than the naive strategy. Although both the approaches converge to the same optimal steady-state solution, dual-based *dFoS* has slower transients (especially around t =15min) due to the chosen controller tuning parameters. This resulted in a slightly smaller cumulative profit $\sum J_{diff}$ compared to HRTO.

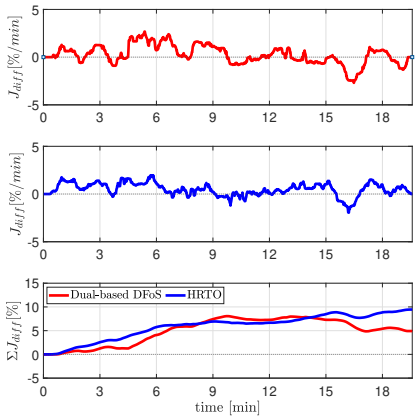


Figure 5.13: The average profit from the Experimental Lab Rig. The bottom subplot shows the cumulative average profit

5.4 Chapter Summary

In this work, we have done experiments to validate the dual-based *dFoS* method. Based on the experiments we can conclude that

- The dual-based *dFoS* is able to provide the same asymptotic optimal performance as HRTO. The transient behaviour is slightly different which is affected by the choice of the controller tuning parameters.
- In the dual-based *dFoS* framework, it is necessary to consider the timescale separation between the gradient and constraint controllers. If the master/central constraint controller is tuned to be in the same timescale as the gradient controllers, then it can lead to instability or oscillatory behavior. However, if the master/central constraint controller is tuned to be too slow, then the convergence to the optimal steady-state can be too slow. This was also seen in the presented experiment results, where dual-based *dFoS* converges slightly slower than the HRTO following some disturbances.

As future work, we suggest the following,

- In this experiment, the shared resource constraint was on the inputs, leading to linear coupling constraints. Since the experimental rig does not have topside separation of liquid and gas, we are not able to include the total produced gas capacity constraint, as done in [41]. Validating the dual-based *dFoS* for nonlienar coupling constraints such as in [41] could be a valuable future research.
- We also used integral controllers in the dual-based *dFoS* framework for its simplicity. Another future research direction could be to consider more advanced controllers for the gradient and master/constraint controllers in the dual-based *dFoS* framework.

Chapter 6

Dual-based DFoS Framework with Single Input Override and Its Systematic Pairing

To be able to minimize constraint violations, as well as dealing with non-performing estimators, while also incorporating automatic active constraint switching within distributed feedback-optimizing system framework, this chapter suggests and describe how Dual-based Distributed Feedback-optimizing system with single input override constraint control and its systematic pairing can be an effective solution. This chapter is based on the work in [47, 88]:

6.1 Introduction

In Chapter 3, we introduced the *primal-dual feedback-optimizing control* with an override feature to address two primary objectives: direct constraint control and fast time scale constraints control. This approach can be further categorized into single-input override and multi-input override.

The first approach, single-input override, is typically preferred when dealing with multiple critical constraints and limited available inputs/manipulated variables (MVs). It excels in minimizing dynamic constraint violations but may have slower convergence (longer transient) when reaching optimal steady-state from the opposite direction. For example, in the case of switching from constrained to unconstrained case or from tight constrained to more relax constrained case. Inappropriate pairing between a chosen input (MV) and a critical constraint can lead to more profit loss. However, finding a systematic procedure for this pairing in interactive systems can be challenging.

Fortunately, Chapter 4 demonstrated a potential strategy for transforming an interactive system into a non-interactive one. Assuming effective performance of the consensus controllers, a systematic pairing procedure can be carried out on a non-interactive system, which offers analytical simplicity. Therefore, this chapter outlines a systematic pairing procedure designed for the dual-based *distributed feedback-optimizing system* framework

with single input/MV override constraint controllers, along with a demonstration of its performance in a subsea oil production network. Meanwhile, the second approach, multi-input override, is discussed in Chapter 7.

6.2 Problem Statement

Consider the following steady-state optimization problem of N different subsystems.

$$\min_{\mathbf{u}} \quad J(\mathbf{u},\mathbf{d}) = \sum_{i=1}^{N} J_i(\mathbf{u}_i,\mathbf{d}_i) \tag{6.1a}$$

$$\text{s.t.} \quad g(\mathbf{u},\mathbf{d}) \leq 0 \tag{6.1b}$$

where $\mathbf{u}_i \in \mathbb{R}^{n_{\mathbf{u}_i}}$ denotes the inputs/MVs for subsystem i, $n_{\mathbf{u}_i}$ is the number of inputs/MVs in subsystem i, and $\mathbf{u} = \begin{bmatrix}\mathbf{u}_1 & \dots & \mathbf{u}_N\end{bmatrix}^\top$, $\mathbf{d}_i \in \mathbb{R}^{n_{\mathbf{d}_i}}$ denotes the disturbances for subsystem i, $n_{\mathbf{d}_i}$ is the number of disturbances in subsystem i, and $\mathbf{d} = \begin{bmatrix}\mathbf{d}_1 & \dots & \mathbf{d}_N\end{bmatrix}^\top$, $J_i : \mathbb{R}^{n_{\mathbf{u}_i}} \times \mathbb{R}^{n_{\mathbf{d}_i}} \to \mathbb{R}$ is a function that denotes the local objective of subsystem i, $g : \mathbb{R}^{n_{\mathbf{u}}} \times \mathbb{R}^{n_{\mathbf{d}}} \to \mathbb{R}^{n_g}$ is a function that denotes the inequality (shared) constraints. n_g is the number of constraints.

The Lagrangian function of problem (6.1) is

$$\mathcal{L}(\mathbf{u},\mathbf{d},\lambda) = \sum_{i=1}^{N} J_i(\mathbf{u}_i,\mathbf{d}_i) + \lambda^\top g(\mathbf{u},\mathbf{d})$$

where $\lambda \in \mathbb{R}^{n_g}$ is the shadow price of the (shared) resource constraints.

To achieve system-wide steady-state optimal operation while adapting to changing active constraints, one viable approach is the dual-based *distributed feedback-optimizing system* framework with single input/MV override constraint controllers. This method employs a master/central constraint controller operating on a slow timescale. However, it suffers from suboptimal performance due to certain practical challenges. For instance, disturbances may occur too rapidly for the master/central constraint controller to handle, or numerical solver-based local subsystems might not update their constrained variables in a timely manner, leading to constraint violations (at least during transient). To address these issues and minimize performance losses during transients, a systematic pairing procedure is crucial to determine the pairing of a manipulated variable (MV) with a constrained variable. This procedure is essential for the selection of the MV in the dual-based *distributed feedback-optimizing system* framework with single input/MV override constraint controller proposed by [47].

When disturbances occur much more frequently than the master/central constraint controller can effectively respond, the proposed solution can be considered a near-optimal strategy.

6.3 Systematic Pairing for Single Input Override Controller

To pair the constrained variables with a good MV, we propose a pairing procedure based on MV's sensitivities to its local disturbances with the following assumptions:

Assumption 6.1: Manipulated and constrained variables conditions

- No saturation issues in the possible MVs, $u^{min} < u_i < u^{max}$.
- Equal value of gain representing relation from any available MVs, i.e, u_i and u_j, to the critical constraint, $G_{u_i} = G_{u_j}$.

Meanwhile, the remaining MVs control their self-optimizing control variables. To describe this proposal, we consider an Indirect control problem formulation as described as the following.

Without loss of generality, consider a case with two MVs, denoted as $u_1 \in \mathbb{R}$ and $u_2 \in \mathbb{R}$. Our objective is to control the gradient of the Lagrange function with respect to its input, denoted as $\mathcal{L}_{\mathbf{u}}(\lambda, \mathbf{u}) \in \mathbb{R}^{2\times 1}$, and the active constrained variable, denoted as $g(\mathbf{u})$. In this context, λ serves as the Lagrange multiplier for constraint function $g(\mathbf{u})$, and $\mathbf{u} = \begin{bmatrix} u_1 & u_2 \end{bmatrix}^\top$. The vector $\mathcal{L}_{\mathbf{u}}(\lambda, \mathbf{u})$ comprises $\mathcal{L}_{u_1}(\lambda, u_1) \in \mathbb{R}$ and $\mathcal{L}_{u_2}(\lambda, u_2) \in \mathbb{R}$.

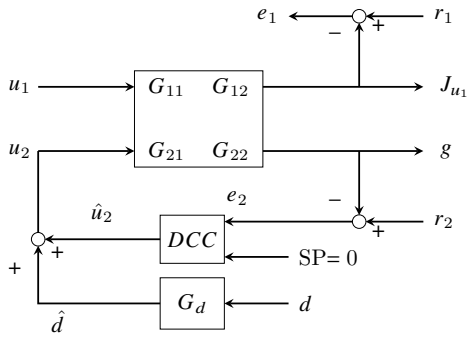


Figure 6.1: Indirect control problem formulation for systematic pairing

Suppose our aim is to control the constrained variables tightly with u_2, considering the influence of a disturbance vector, $\mathbf{d} \in \mathbb{R}^{2\times 1}$, on input u_2. This disturbance can result from either local disturbances in subsystem 2 or changes in u_2 due to setpoint adjustments, often originating from changes in subsystem 1. As λ remains relatively constant (due to the non-performing or very slow time scale constraint control) and $g_{u_1}(u_1)$ is typically constant (e.g., resource allocation), we primarily aim to control $J_{u_1}(u_1)$. This problem can be expressed as an indirect control problem with the following set of equations:

$$J_{u_1}(u_1) = G_{11}u_1 + G_{12}u_2 \tag{6.2a}$$
$$g(\mathbf{u}) = G_{21}u_1 + G_{22}u_2 \tag{6.2b}$$
$$u_2 = G_d d + \hat{u}_2 \tag{6.2c}$$

In this formulation, G_{11} represents the gain from u_1 to $J_{u_1}(u_1)$, G_{12} is the gain from u_2 to $J_{u_1}(u_1)$, G_{21} is the gain from u_1 to $g(\mathbf{u})$, G_{22} is the gain from u_2 to $g(\mathbf{u})$, and G_d is the disturbance gain affecting u_2.

The objective, as shown in Figure 6.1, is to tightly control $g(\mathbf{u})$ to the reference r_2 directly using an override/direct constraint controller (DCC). Simultaneously, we aim to

find u_2 such that u_2 indirectly controls $J_{u_1}(u_1)$ to reference r_1 or through pairing with $g(\mathbf{u})$. This control structure offers better control of $J_{u_1}(u_1)$ compared to other possible structures.

We assume that G_{22} is square and invertible. Otherwise, the solution can be approximated using the pseudoinverse. By rearranging Equation 6.2 and assuming perfect control of $g(\mathbf{u}) \approx r_2$, we can estimate that:

$$J_{u_1}(u_1) \approx G_{12} G_{22}^{-1} r_2$$

which means that any small changes in r_2 leads to small changes in $J_{u_1}(u_1)$ with linear relation. Hence, we must choose r_2 such that:

$$r_2 \approx G_{22} G_{12}^{-1} r_1$$

According to [46], $G_{12}G_{22}^{-1}$ should ideally be small. This often implies a preference for pairings that yield the largest G_{22} value, where $G_{22} = \nabla_{u_2} g(\mathbf{u})$. However, as per Assumption 6.1, this approach may not be sufficient.

The presented formulation highlights the need to also consider small G_{12} values in addition to large G_{22} values, especially when dealing with faster disturbances or a non-performing upper layer in the dual-based distributed feedback-optimizing control system *framework with override constraint controller.*

Selecting based on G_{12} complements common rules, constituting a near-optimal strategy, particularly during transient, for the dual-based *distributed feedback-optimizing control system* framework with an override constraint controller [47].

By defining $\hat{d} = G_d d$, G_{12} can be approximated as $\frac{\Delta J_{u_1}}{\Delta u_2} = \frac{\Delta J_{u_1}}{\Delta(\hat{d}+\hat{u}_2)}$. Given that keeping $\hat{u}_2$ at the same value controls $g(\mathbf{u})$, any change in $\hat{d}$ can signify a shift. Referring to Equation 6.2c, $G_{12} \approx \frac{\Delta J_{u_1}}{\Delta \hat{d}}$. Moreover, assuming the stationary point aligns with the local optimum and considering that J_{u_1} is controlled by u_1, and J_{u_2} remains uncontrolled, any disturbance to J_{u_1} leads to a shift in ΔJ_{u_2} (indicating $\mathcal{L}_{u_2}$ drifts from 0). Consequently, any disturbance on J_{u_1} leads to a total profit loss ΔJ. Therefore, we can estimate G_{12} as follows:

$$G_{12} \approx \frac{\Delta J}{\Delta \hat{d}}$$

6.4 Numerical Examples

6.4.1 Systematic Pairing Analysis

We apply the rules in a subsea gas-lifted oil production optimization problem with a fixed gas lift compressor as described in [47], where the power consumption of the compressor is modelled as a linear function:

$$Pow_{gl} = \theta \sum_{i=1}^{N} w_{gl,i}$$

where θ is a function of a fixed ratio of outlet and inlet pressure of the compressor. Further, N is the total number of wells, and $w_{gl,i}$ is the gas-lift rate injected to well i. The case

study involves a subsea gas-lifted oil production well network consisting of two wells and each well is operated locally (Figure 6.2 illustrates this case and Table 6.1 presents the model parameters). This means both control and parameter estimation in the optimization framework are performed locally (*distributed feedback-optimizing system*). The optimization goal is to maximize oil production income while minimizing gas lift costs. The optimization problem can be expressed as follows:

$$\min_{\mathbf{w}_{gl}} \quad \sum_{i=1}^{N} \left(-p_{o,i} w_{po,i} + p_{gl,i} w_{gl,i}\right) \tag{6.3a}$$

$$\text{s.t.} \qquad \mathbf{f}\left(\mathbf{x}, \mathbf{w}_{gl}, \mathbf{d}\right) = 0, \tag{6.3b}$$

$$\mathbf{g}\left(\mathbf{x}, \mathbf{w}_{gl}, \mathbf{d}\right) \leq 0, \tag{6.3c}$$

$$\mathbf{g_s}\left(\mathbf{x}, \mathbf{w}_{gl}, \mathbf{d}\right) = Pow_{gl} - Pow_{gl}^{max} \leq 0 \tag{6.3d}$$

where $p_{o,i}$, $p_{gl,i}$ and $w_{po,i}$ are the price of produced oil, the cost of gas-lift, and the produced oil rate of well i, respectively. Pow_{gl} is the total power consumed by the fixed compressor to inject the total gas-lift rate i, and Pow_{gl}^{max} is the maximum available power. The vector $\mathbf{x} \in \mathbb{R}^{n_\mathbf{x}}$, and $\mathbf{d} \in \mathbb{R}^{n_\mathbf{d}}$ are the vectors of states, and disturbance (i.e., gas-oil-ratio) for the entire system. $n_\mathbf{x}$ is the number of states. $\mathbf{w}_{gl} \in \mathbb{R}^{n_{\mathbf{w}_{gl}}}$ is the vector of inputs for the entire system, where $\mathbf{w}_{gl} = \begin{bmatrix} w_{gl,1} & \dots & w_{gl,N} \end{bmatrix}^\top$. Constraint (6.3b) and (6.3c) represent model and physical constraints, respectively. We assume that Constraint (6.3d) is locally managed to maintain the focus of the discussion. Eq. (6.3a) is additively separable, and eq. (6.3d) is a linear and hard constraint. This setup utilizes a fixed-efficiency gas lift compressor to supply the total gas lift rate.

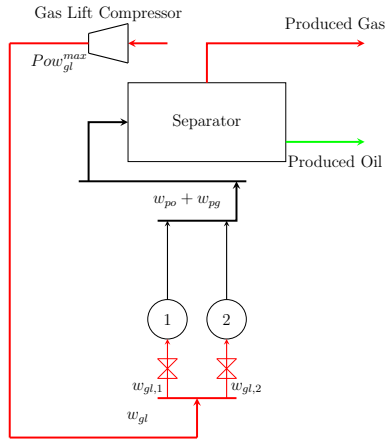


Figure 6.2: Field Illustration

We perform simulations where the Lagrange multiplier, λ, remains non-updated. We present numerical results for the proposed pairing procedure or near-optimal performance strategy (Structure 1) and compare its performance to the asynchronous protocol (Structure 0), in which local controllers continuously maintain the gradient of the Lagrange function

Table 6.1: List of wells and gas lift compressor parameters in the simulation.

Par. [units]	Well 1	Well 2	Gas lift compressor
— See Table 4.5 —			-
p_m [bar]	20		-
θ [MWs/kg]	-		3674

with respect to the input at zero. We also provide results for an alternative structure (Structure 2). For clarity, Table 6.2 present the pairing for each structure.

Table 6.2: Pairing in each structure

Structure	Pairing 1	Pairing 2
0	$w_{gl,1} \leftrightarrow \mathcal{L}_{w_{gl,1}}$	$w_{gl,2} \leftrightarrow \mathcal{L}_{w_{gl,2}}$
1	$w_{gl,1} \leftrightarrow \mathcal{L}_{w_{gl,1}}$	$w_{gl,2} \leftrightarrow \mathbf{g_s}$
2	$w_{gl,1} \leftrightarrow \mathbf{g_s}$	$w_{gl,2} \leftrightarrow \mathcal{L}_{w_{gl,2}}$

We solve the steady-state optimization problem (6.3) to determine the optimal cost for various disturbance cases, assuming a maximum historical disturbance error of ±5%. The profit loss, denoted as $\Delta J_j = J_j - J^\star$, is computed as the difference between the steady-state cost of each structure (indexed by $j \in 0, 1, 2$) and the optimal cost.

First, we simulate the largest possible error case for Structure 0, which experiences sequential disturbances starting with $GOR_1 + 5\%$, $GOR_1 - 5\%$, $GOR_2 + 5\%$, $GOR_2 - 5\%$, $Pow_{gl}^{max} + 5\%$, and finally $Pow_{gl}^{max} - 5\%$. Figure 6.3 illustrates that Structure 0 fails to meet steady-state constraints when GOR_1, GOR_2, or Pow_{gl}^{max} decreases by 5% (time windows 18-32 hr, 48-62 hr, and 78-90 hr). This validates the necessity of a near-optimal strategy within the dual-based *distributed feedback-optimizing system* framework.

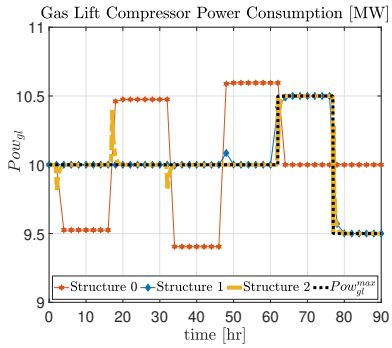


Figure 6.3: Steady-state constraint satisfaction

As discussed in Section 6.3, the first rule is to pair the input and active constraint with the largest $G_{22,j} = \nabla_{w_{gl,j}} \mathbf{g}s\,(\mathbf{x}, \mathbf{w}gl, \mathbf{d})$. We find that $G_{22,1} = 3.6740$ and $G_{22,2} = 3.6740$, which confirms the assumption of equal constraint-MVs gain and emphasizes the need for an additional rule to achieve economically oriented results.

The second rule involves pairing the input and active constraint with the smallest $G_{12,j}$, estimated by calculating $\frac{\Delta J_j}{\Delta GOR_j}$ using the finite difference method. The results indicate that the smallest $G_{12,1}$ is 1.4441, and the smallest $G_{12,2}$ is 1.4642. Following the method outlined in Section 6.3, these findings suggest that Structure 1, pairing the active constraint with $w_{gl,2}$, is the most economically favorable.

Figure 6.4 compares profit loss between Structure 1 and Structure 2, demonstrating that Structure 1 consistently minimizes steady-state loss more effectively under extreme disturbances. Table 6.3 presents the steady-state profit loss for 24 hours in various extreme disturbance scenarios.

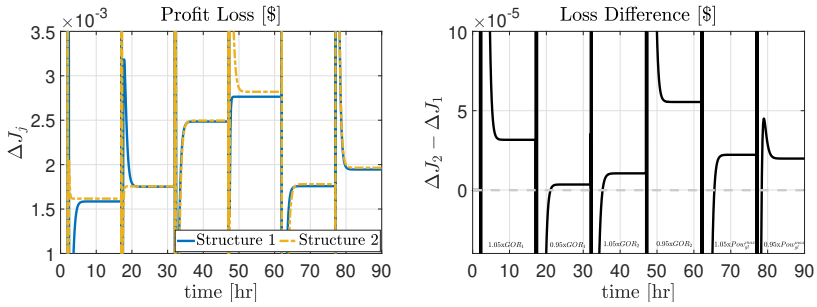


Figure 6.4: Profit loss comparison (per *second*) (left) and loss difference (per *second*) between Structure 1 and 2 (right).

Table 6.3: Steady-state profit loss per *day*

Structure	0	1	2
$GOR_1 + 5\%$	630.7200	136.3423	139.7161
$GOR_1 - 5\%$	sscv*	151.6720	151.6730
$GOR_2 + 5\%$	751.6800	214.7048	215.0000
$GOR_2 - 5\%$	sscv*	238.8445	243.8771
$Pow_{gl}^{max} + 5\%$	656.6400	151.4331	153.3618
$Pow_{gl}^{max} - 5\%$	sscv*	168.2676	169.9576

sscv* : steady-state constraint violation.

6.4.2 Implementation in Subsea Oil Production Network

In the context of the same case study discussed in the previous section, this section presents a comparative analysis of the simulation results obtained from both *dFoS* frameworks with single-input override. Refer to Figure 6.5 for structure 1 and Figure 6.6 for structure 2. Please note that $y_{m,i}$ refers to the set of real-time measurements managed locally by subsystem i. The diagram illustrates the physical system within the grey boxes, with computation blocks denoted by white boxes. These computation blocks are further categorized into three levels of timescales (from fast to slow), indicated by light green, light blue, and light red boxes. For a more focus comparison with the standard primal-dual or *dFoS* framework [41, 40], readers can refer to Chapter 3.

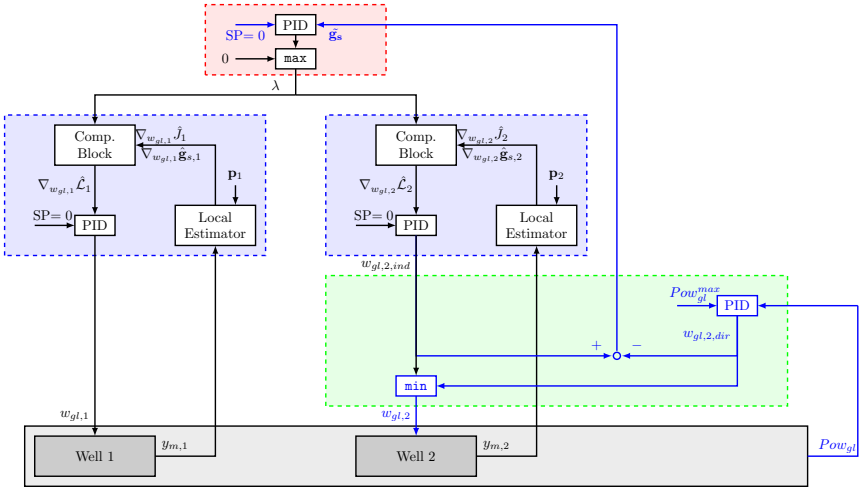


Figure 6.5: Dual-based *dFoS* framework with single input override - Structure 1

Based on our systematic pairing analysis, we found that Structure 1, depicted in Figure 6.5, is the preferred choice for minimizing profit loss during transient. Our original constraint $\mathbf{g_s} \leq 0$ has been examined using step response, revealing that $\frac{d\mathbf{g_s}}{dw_{gl,2}} > 0$, indicating that a small value of $w_{gl,2}$ is good in terms of satisfying the constraint. To implement this preference, we utilize a `min` selector, denoted as:

$$w_{gl,2} = \texttt{min}\left(w_{gl,2,dir}, w_{gl,2,ind}\right)$$

where $w_{gl,2,dir}$ is the manipulated variable computed by the override/direct constraint controller, and $w_{gl,2,ind}$ is the primal manipulated variable set by the gradient controller. It is important to note that, in the optimal steady state, we must ensure $w_{gl,2,dir} \geq w_{gl,2,ind}$ or equivalently $\tilde{\mathbf{g}}_s = w_{gl,2,ind} - w_{gl,2,dir} \leq \mathbf{0}$—a constraint controlled by the master controller, as indicated in the light red box in Figure 6.5. This constraint is defined as *auxiliary* constraint.

For further insight into this case study, please refer to Table 6.4, which shows the controlled variables, setpoints, and manipulated variables. Additionally, we have employed the same estimation method used in [41] to calculate the steady-state cost and constraint gradient, denoted as $\nabla_{w_{gl,i}} \hat{\mathbf{J}}$ and $\nabla_{w_{gl,i}} \hat{\mathbf{g}}_s$, respectively.

Table 6.4: Controlled Variables, Setpoints, and Manipulated Variables

Well	CV	CV^{sp}	Calc. MV	Phys. MV
1 (indirect)	$CV_{1,ind} = \nabla_{w_{gl,1}} \mathcal{L}_1$	0	$w_{gl,1,ind}$	$w_{gl,1}$
2 (indirect)	$CV_{2,ind} = \nabla_{w_{gl,2}} \mathcal{L}_2$	0	$w_{gl,2,ind}$	$w_{gl,2}$
2 (direct)	$CV_{2,dir} = Pow_{gl}$	Pow_{gl}^{max}	$w_{gl,2,dir}$	$w_{gl,2}$

The concept of the *auxiliary* constraint involves adjusting the shadow price λ to ensure that, over time, the manipulated variable's (MV) value computed by the direct/override

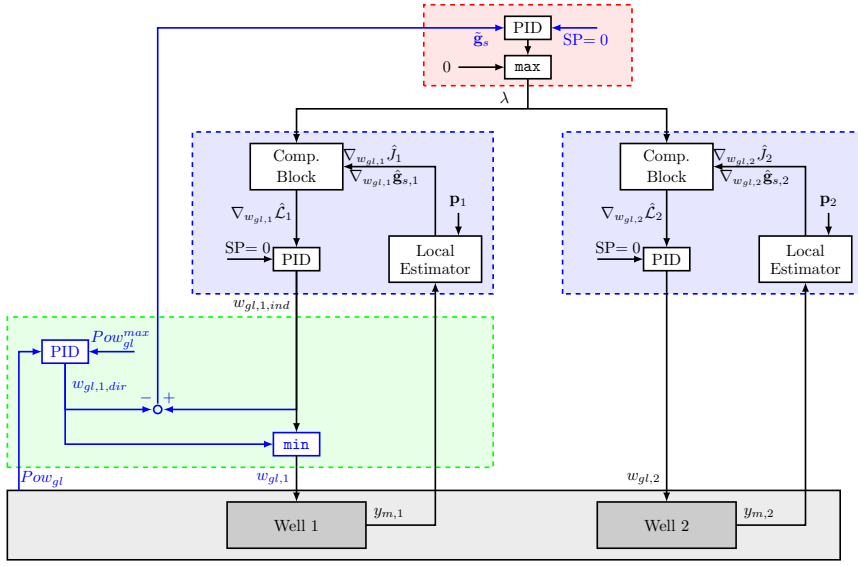


Figure 6.6: Dual-based *dFoS* framework with single input override - Structure 2

constraint control aligns with the steady-state optimal primal value determined by the gradient controller when the constraint is active. To calculate the appropriate λ, a master *auxiliary* constraint controller with a `max` selector can be employed. This selector, which switches to 0 when the constraint is no longer active. The anti-windup serves to minimize constant fluctuations in λ when there are active constraint region transitions.

Thus, this selector switches between 0 and the computed shadow price $\hat{\lambda}$. The shadow price $\hat{\lambda}$ at iteration k can be expressed as follows:

$$\hat{\lambda} = \lambda^k + K_P \tilde{g}_s^k + \sum_{\tau=k-1}^{k} \left(K_I \tilde{g}_s^\tau + K_{AW} \left(\lambda - \hat{\lambda} \right)^\tau \right) \tag{6.4}$$

In this equation, K_P, K_I, and K_{AW} represent the proportional, integral, and anti-wind-up gain, respectively.

PI controllers in this study are tuned using the SIMC tuning method as introduced by [52]. The controllers have a sampling time of 1 second. The master auxiliary constraint controller operates 5 times slower than the local controllers.

Meanwhile, the override constraint controller runs at a significantly higher rate, performing tasks five times faster than the local controllers. For detailed information about the controllers and their corresponding tuning parameters, please refer to Table 6.5. It is important to note that these controllers are designed without anti-windup mechanisms, as this study focuses on scenarios where constraints are consistently active. In general cases, the inclusion of anti-windup measures is necessary.

Figure 6.7 demonstrates the dynamic constraint violations for Structure 0, Structure 1 and Structure 2. Both structure 1 and 2 significantly minimize violations compared to the standard *dFoS* framework (structure 0). Towards the simulation's time, the structure

Table 6.5: Parameters used in PID controllers shown in Figure 6.5- 6.6

Tuning Par.	Local Grad. Controller 1	Override Controller 1	Local Grad. Controller 2	Override Predictor 2	Aux. Const. Controller
K_P	8.5696	–	8.2642	–	-0.0174
K_I	0.0129	0.0068	0.0130	0.0068	$-4.3543e-05$

0 requires an additional 6310.3 MW of power. In contrast, structure 1 only necessitates 122.2 MW, and structure 2 requires 140.6 MW.

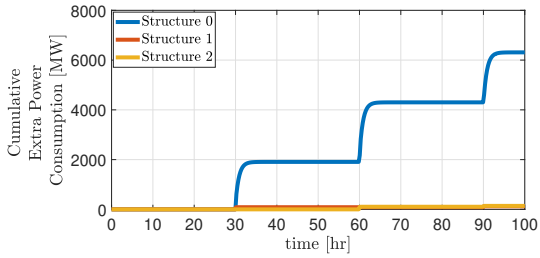


Figure 6.7: Cumulative dynamic violation over the simulation time.

The transient time can significantly longer, depending on the scenario, particularly when transitioning from a tight constrained case to a more relaxed one. This occurs, for instance, with an increase in Gas-Oil Ratio (GOR) or a relaxation of power limits. Figure 6.8 illustrates that Structure 1 is closer to the constraint limit (when the value is negative), while Structure 2 is closer to the constraint limit (when the value is positive). Notably, after disturbances at times $t = 10$ (GOR 1 increases), $t = 45$ (both GOR 1 and 2 increase), and $t = 75$ hours (maximum power limit increases), neither Structure 1 nor Structure 2 have reached a steady state before the subsequent disturbance. This aligns with the practical assumption discussed in Section 6.3. Hence, minimizing loss during transients is importance, and finding the best pairing is crucial.

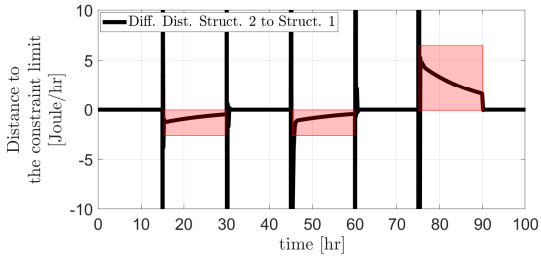


Figure 6.8: Difference between structure 2 and 1 in distance of constrained variable to the limit.

In summary, Structure 1 incurs an average monthly loss of USD 95,381, while Structure 2 records an average monthly loss of USD 95,770. Choosing Structure 1 results in a reduction of USD 389 in monthly losses on average, representing a 0.41% decrease in

losses, which is quite nice for an almost identical two wells network, and with no extra cost. This outcome serves as a numerical validation of the approach outlined in Section 6.3.

6.5 Chapter Summary

This chapter, as introduced by [89], presents a systematic pairing approach for single-input override constraint controllers within the *dFoS* framework. The proposal's validity is numerically affirmed through the implementation of the *dFoS* framework with single-input override in a subsea oil production network case study. This substantiates the feasibility of integrating override controllers and *auxiliary* constraint controllers into the *dFoS* framework, as proposed in [47].

Assessment Table:

Let S-X be the proposed framework of this chapter, the parameter we assess and compare are as described in section 2.5. Table 6.6 summarizes the assessment, where the

Table 6.6: Summary of Chapter 6

	S-IV	S-VI	S-IX	S-X
(a)	Yes	Yes	Yes	Yes
(b)	Yes	Yes	Yes	Yes
(c)	Yes	Yes	Yes	Yes
(d)	No[1]	No[1]	No[1]	Yes
(e)	No	Yes	Yes	Yes
(f)	No (Medium)	Yes	Yes	Yes
(g)	Yes	No[2]	Yes	Yes
(h)	No	Yes	Yes	Yes
(i)	No	No	No	No
(j)	Yes	No	No	No

"Yes": the approach has satisfied the success parameter
"Yes": the approach has satisfied the success parameter in this Chapter
"No": the approach does not satisfied the success parameter
"No": does not satisfied the success parameter, and will be addressed in this thesis.
[1]: will be addressed in Chapter 3
[2]: will be addressed in Chapter 4
"No (High)": does not satisfied the success parameter (f) at all (shown in Chapter 2).
"No (Medium)": has partly satisfied the success parameter (f) better than "No (High)".

approaches are:

- S-IV: Dual-based hybrid RTO
- S-VI: Primal dual FOC
- S-IX: Dual-based Distributed Feedback-Optimizing System
- S-X: Dual-based Distributed Feedback-Optimizing System with Override

and the parameters we assess and compare are:

- (a): Is steady-state optimal achieved?
- (b): Is constraint controlled *transparently* (see Definition 2.1)?
- (c): Is flexible for active constraint switching?

- (d): Is constraint controlled *directly* (on fast time scale)?
- (e): Is applicable for less than twice differentiable Lagrange function?
- (f): Is numerical solver avoidable (computation time)?
- (g): Is it recommended for complex and large system?
- (h): Is input filter (or additional setpoint controller) avoidable?
- (i): Is parameters and states dynamic estimator avoidable?
- (j): Is gradient estimator avoidable?

Future works: In future works, an intriguing avenue of research involves the development of a control structure featuring multi-input override controllers equipped by auxiliary constraint controllers. This study will be the focus of Chapter 7. Additionally, Chapter 9 will show the experimental validation of this approach.

Chapter 7

Procedures for Dual-based DFoS Framework with Multi-Input Override

In cases where parallel systems, such as gas-lift oil production optimization, are involved, it is often possible to utilize multiple MVs concurrently to address important constraints. To be able to effectively exploit the potential of Override Constraint Control within the Dual-based dFoS *Framework, this chapter suggests and describes how to construct multiple MVs (multi-input). This chapter is based on the work in [48].*

7.1 Introduction

As introduced in Chapter 6, the single-input override approach is commonly favored when addressing scenarios involving multiple critical constraints and limited available inputs or manipulated variables (MVs). However, it is also common in the process industry to encounter parallel systems where an abundance of available inputs/MVs collaboratively contribute to the control of a critical constraint.

The following sections of this Chapter is dedicated to explore various techniques within the dual-based distributed feedback-optimizing system (*dFoS*) framework, focusing on the implementation of multi-input or MV override constraint controllers. To illustrate these concepts, we consider a practical case study involving a subsea oil production network comprising parallel gas-lifted oil production wells.

7.2 Problem Statement

Chapter 3 introduced the concept of an *auxiliary* input constraint and implicit override as a solution to address input saturation issues. Identifying inputs prone to saturation can be challenging, especially in the presence of unpredictable disturbances like those from geological structures. Even if potential input constraints are identified, once saturation occurs, the affected input temporarily loses its ability to control critical constraints, potentially causing increased dynamic constraint violations. Furthermore, practical im-

plementations often face the constraint of limited actuator rates, contributing to further challenges in dealing with dynamic constraints as illustrated in Figure 7.1. This underscores the need for a more robust control structure, as relying on a single input to handle critical constraints has inherent limitations.

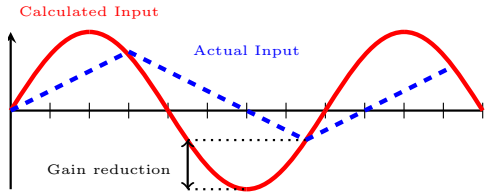


Figure 7.1: Calculated and actual input in the presence of input rate saturation, adopted from [90].

Therefore, it is both practical and reasonable to design and implement a multi-input override within the dual-based *dFoS* framework. This approach provides enhanced control flexibility, allowing multiple inputs to contribute to the control of critical constraints. In dynamic and uncertain environments, such a multi-input override strategy offers a more resilient and adaptable solution, mitigating the impact of input saturation and rate constraints on the overall control performance.

7.3 Proposed Approach(es)

To address the issue outlined in Section 7.2, we introduce a multi-input override constraint control structure within the dual-based *dFoS* framework. The block diagram visualizing this proposal is shown in Figure 7.2.

In the previous Chapter 6, we discussed single-input override constraint control [47], where a single input is designated to regulate an active critical constraint. In the proposed approach, multiple inputs jointly control the constraint directly on fast time scale. This modification is highlighted with blue arrows and blocks in Figure 7.2, and we discuss this topic in this chapter. Stacked white blocks on a light blue background represent a number of independent subsystems which are the result of decomposition. Other aspects depicted in the figure have been previously explained in earlier chapters.

7.3.1 Transform multi-input into single-input

To implement override constraint control, we must carefully select one combined input vector ($\mathbf{u}^c$) for each critical constraint, as described by the equation:

$$\mathbf{u}^c = \mathbf{H}^\top \mathbf{u}_{ind} \tag{7.1}$$

where $\mathbf{u}_{ind} \in \mathbb{R}^{n_\mathbf{u}}$ denotes the vector of all inputs provided by gradient controllers, denoted with a light blue background.

In the scenario where only one critical constraint requires direct or tight control, we simply choose a single input, resulting in $\mathbf{H}$ being a row vector. This vector is primarily filled with zeros except for a single non-zero entry corresponding to the selected input. However, in more complex cases involving multiple critical constraints that need simultaneous tight control, we introduce a selection matrix $\mathbf{H}$. This matrix, denoted by $\mathbf{H} \in \mathbb{R}^{n_\mathbf{u} \times n_{\mathbf{g}_\mathbb{A}}}$, where $n_{\mathbf{g}_\mathbb{A}}$ is the number of critical constraints that we want to control

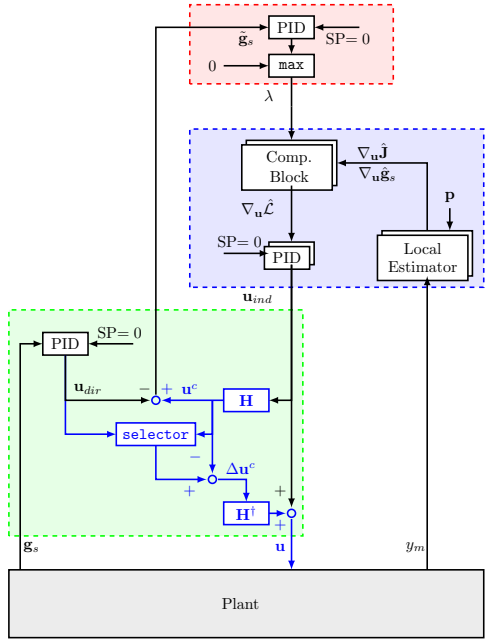


Figure 7.2: Block diagram of multi-input override control within the dual-based *dFoS* framework.

tightly when they are active simultaneously. For instance, we might choose the following configuration:

$$\mathbf{H}^\top = \begin{bmatrix} h_{11} & h_{21} & 0 & 0 \\ 0 & 0 & h_{32} & h_{42} \end{bmatrix} \tag{7.2}$$

In this example, the first two inputs are allocated to jointly control constraint g_1, while the last two inputs are assigned to constraint g_2. Therefore, the selection matrix $\mathbf{H}$ offers a versatile configuration where a constraint can be directly or tightly controlled by multiple inputs.

7.3.2 Various procedures constructing selection matrix H

Option 1: Gain matrix as selection matrix

A rational approach is to prioritize inputs with a significant effect on the constraints. Therefore, a straightforward choice is to consider:

$$\mathbf{H} = \nabla_{\mathbf{u}}\mathbf{g}_{\mathbb{A}} \tag{7.3}$$

where $\nabla_{\mathbf{u}}\mathbf{g}_{\mathbb{A}}$ represents the gradient of constraints that are active simultaneously with respect to the input. However, it is essential to assign less weight to inputs that are operating close to their constraints or have a significant effective delay to the corresponding constraint. For instance, we may opt for the following configuration:

$$\mathbf{H}^\top = \begin{bmatrix} \nabla_{u_1} g_1 & \nabla_{u_2} g_1 & 0 & 0 \\ 0 & 0 & \nabla_{u_3} g_2 & \nabla_{u_4} g_2 \end{bmatrix} \tag{7.4}$$

Option 2: Gain ratio matrix as selection matrix

Engineers commonly prioritize handling a critical constraint by selecting a specific input to emphasize. To achieve this, we may normalize other inputs relative to the gain of the chosen prioritized input. For instance, we might designate input u_1 as the top-priority input for managing constraint g_1 and input u_4 as the primary input for handling constraint g_2. As a result, the selection matrix takes the following form:

$$\mathbf{H}^\top = \begin{bmatrix} \frac{\nabla_{u_1} g_1}{\nabla_{u_1} g_1} & \frac{\nabla_{u_2} g_1}{\nabla_{u_1} g_1} & 0 & 0 \\ 0 & 0 & \frac{\nabla_{u_3} g_2}{\nabla_{u_4} g_2} & \frac{\nabla_{u_4} g_2}{\nabla_{u_4} g_2} \end{bmatrix} \tag{7.5}$$

It is expected that this approach ensures the prioritized inputs effectively address the critical constraints while maintaining appropriate interrelations between the inputs.

Option 3: Disturbance sensitivity based matrix as selection matrix

The matrix formulation is devised to regulate the constraint as closely as possible to its limit by compensating for gaps caused by disturbances. Given that the entire system is interconnected through the coupling constraint, disturbances, regardless of their origin, impact other subsystems and, consequently, affect the constraint. Therefore, we can assume that the number of constraints is equal to the number of disturbances ($n_\mathbf{g} = n_\mathbf{d}$), and the source of disturbance can arise from any subsystems.

Let $\boldsymbol{\zeta} = \begin{bmatrix} \mathbf{u} & \boldsymbol{\lambda} \end{bmatrix}^\top$. In the active constraint condition, optimal operation is achieved as follows:

$$\mathcal{V}(\boldsymbol{\zeta}, \mathbf{d}) = \begin{bmatrix} \nabla_\mathbf{u} \mathcal{L}(\mathbf{u}, \boldsymbol{\lambda}, \mathbf{d}) & \mathbf{g}(\mathbf{u}, \mathbf{d}) \end{bmatrix}^\top = \begin{bmatrix} \mathbf{0} & \mathbf{0} \end{bmatrix} \tag{7.6}$$

The total derivative of Equation (7.6) is given by:

$$\nabla_{\boldsymbol{\zeta}} \mathcal{V} d\boldsymbol{\zeta} + \nabla_\mathbf{d} \mathcal{V} d\mathbf{d} = \mathbf{0} \tag{7.7}$$

where

$$\nabla_{\boldsymbol{\zeta}} \mathcal{V} = \begin{bmatrix} \nabla_{\mathbf{uu}} \mathcal{L}(\mathbf{u}, \boldsymbol{\lambda}, \mathbf{d}) & \nabla_\mathbf{u} \mathbf{g}(\mathbf{u}, \mathbf{d}) \\ \nabla_\mathbf{u}^\top \mathbf{g}(\mathbf{u}, \mathbf{d}) & \mathbf{0} \end{bmatrix}$$

and

$$\nabla_\mathbf{d} \mathcal{V} = \begin{bmatrix} \nabla_{\mathbf{ud}} \mathcal{L}(\mathbf{u}, \boldsymbol{\lambda}, \mathbf{d}) \\ \nabla_\mathbf{d} \mathbf{g}(\mathbf{u}, \mathbf{d}) \end{bmatrix}$$

Rearranging Equation (7.7), we obtain:

$$\frac{d\boldsymbol{\zeta}}{d\mathbf{d}} = \begin{bmatrix} \frac{d\mathbf{u}}{d\mathbf{d}} \\ \frac{d\boldsymbol{\lambda}}{d\mathbf{d}} \end{bmatrix} = -\left(\nabla_{\boldsymbol{\zeta}} \mathcal{V}^{-1}\right) \nabla_\mathbf{d} \mathcal{V} \tag{7.8}$$

where $\frac{d\boldsymbol{\zeta}}{d\mathbf{d}} \in \mathbb{R}^{(n_\mathbf{u} + n_\mathbf{g}) \times n_\mathbf{g}}$ and $\frac{d\mathbf{u}}{d\mathbf{d}} \in \mathbb{R}^{n_\mathbf{u} \times n_\mathbf{g}}$. Finally, the selection matrix is defined as follows:

$$\mathbf{H} = c. \left| \frac{d\mathbf{u}}{d\mathbf{d}} \right| \tag{7.9}$$

where c is a tuning parameter.

7.3.3 Combining override constraint controllers correction

Given $\mathbf{u}_{ind} \in \mathbb{R}^{n_\mathbf{u}}$, we can calculate the input $\mathbf{u}$ to be implemented on the plant using the equation:

$$\mathbf{u} = \mathbf{u}_{ind} + \left(\mathbf{H}^{\dagger}\right)^{\top} \Delta\mathbf{u}^{c} \tag{7.10}$$

where $\mathbf{H}^{\dagger} \in \mathbb{R}^{n_{\mathbf{g}_\mathbb{A}} \times n_\mathbf{u}}$ is the pseudo inverse of $\mathbf{H}$, and $\Delta\mathbf{u}^{c}$ labels the correction factor computed as follows:

$$\Delta\mathbf{u}^{c} = \left(\min\left(\mathbf{u}_{dir}, \mathbf{H}^{\top}\mathbf{u}_{ind}\right) - \mathbf{H}^{\top}\mathbf{u}_{ind}\right) \tag{7.11}$$

where $\mathbf{u}^{c}_{dir} \in \mathbb{R}^{n_{\mathbf{g}_\mathbb{A}}}$ is the input obtained from override constraint controller, denoted in light green background.

The choice of a `min` selector is based on the assumption that increasing the input brings the constrained variables closer to the constraints. If the response of input-constrained variables is in the opposite direction, a `max` selector would be used instead.

When we have two or more constraints that are never active simultaneously, and we assign all available inputs to tightly control the active one, the selected correction factor can be defined as follows:

$$\Delta\mathbf{u}^{c} = \min\left(\Delta\mathbf{u}^{c}_{g_1}, \Delta\mathbf{u}^{c}_{g_2}\right)$$

where $\Delta\mathbf{u}^{c}_{g_1}$ and $\Delta\mathbf{u}^{c}_{g_2}$ are the correction factors when g_1 and g_2 are active, respectively. The implementation of this switching strategy can be found in [48].

When we assume that two constraints may be active simultaneously, we could split the inputs, i.e., the first two inputs may be used to control the first constraint tightly, and the last two may be used to the second constraint. One possible option of the selection matrix for this formulation is:

$$\mathbf{H}^{\top} = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \end{bmatrix}$$

and it is not necessary to select the correction factor as both of the selection factors are required by the assigned inputs. To the end, this approach provides flexibility in constructing the control structure that can satisfy any possible active constraints region/combination.

7.4 Numerical Examples

7.4.1 Various Multi-input Approaches in Active Constraint Handling

Problem and case description

We apply the proposed approach to a gas-lifted oil production network consisting of six wells ($N = 6$), a setup closely resembling the one utilized in [47]. Each well operates locally, and the oil production from each well, represented as $w_{po,i}$, is regulated through the gas-lift injection rate $w_{gl,i}$ (noted as $\mathbf{u} = \mathbf{w}_{gl}$). The capacity for handling the total exported gas is limited, constrained to a maximum supply of w_{pg}^{max} (linked to constraint function g). Figure 7.3 illustrates this case study. Within this setup, a manipulated variables (MV) are prioritized to control critical constraints, namely, $w_{gl,4} \leftrightarrow g$.

The ultimate objective of the production optimization problem is to optimally allocate the shared gas lift among the wells in a manner that maximizes profit. This objective is

encapsulated in the following optimization problem:

$$\min_{w_{gl,i},\, \forall i \in N} \quad J_N = -\sum_{i \in N} \$_{o,i} w_{po,i} + \sum_{i \in N} \$_{gl,i} w_{gl,i} \tag{7.12a}$$

$$\text{s.t.} \quad g(\mathbf{u}, \mathbf{d}) =: \sum_{i \in N} w_{pg,i} - w_{pg}^{max} \leq 0 \tag{7.12b}$$

where $\$_{o,i}$ and $\$_{gl,i}$ are the oil prices and the cost associated with gas compression for gas-lift purposes, respectively. The total exported gas is denoted as $R_{pg_N} = \sum_{i \in N} w_{pg,i}$.

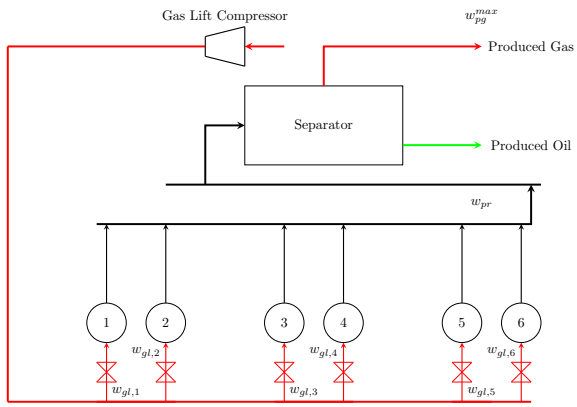


Figure 7.3: A simplified process diagram of a gas-lift case with limited capacity for exported gas.

Furthermore, it is common for valves to be subject to input rate constraints. These constraints are defined as follows:

$$\Delta w_{gl}^{min} \leq \Delta w_{gl,i} \leq \Delta w_{gl}^{max}, \; i = 1, ..., N$$

where, Δw_{gl}^{min} and Δw_{gl}^{max} are the lower and upper bound of the inputs rate, respectively.

Additionally, variations in the gas-oil ratio (GOR), a reservoir property, are considered, which are often associated with disturbances affecting different wells over time. Wells with a high GOR have a lighter fluid column, reducing the need for extensive gas-lift injection compared to wells with a low GOR. Consequently, changes in the GOR have a significant impact on the optimal allocation of gas-lift injection.

Benchmarks

In our simulation-based comparisons, we focus on assessing the performance of various approaches in active constraint handling. To maintain clarity and focus of this work, we consider the following approaches:

- (C1) The standard dual-based *dFoS*, which can achieve steady-state optimal.
- (C2) The dual-based *dFoS* equipped with single-input override constraint control.
- (C3.1, C3.2, C3.3) The dual-based *dFoS* equipped with multi-input override constraint control. This category includes three distinct options for constructing the selection matrix: Option 1, Option 2, and Option 3, respectively

The standard dual-based *dFoS* framework

To address problem (7.12) using the standard dual-based *dFoS* framework (C1), please refer to [41] or Chapter 4.

Single-input override constraint control in the dual-based *dFoS* framework

The single-input approach (C2) is a specific instance of the multi-input approach. In line with the critical constraint configuration detailed earlier, we can represent the pairings in a matrix form by choosing matrix $\mathbf{H}$ as:

$$\mathbf{H}^\top = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix}$$

For the sake of clarity, we also exercise a simulation of single-input approach without any input rate constraint, labelled as C2.0.

Multi-input override constraint control in the dual-based *dFoS* framework

In this scenario, we assume that subsystem/well 1, 2, and 3 have no access to contribute in handling constraint g, and each of them applies the standard dual-based *dFoS* framework. As as results, in multi-input control structure, we choose:

- $\mathbf{H}^\top = \begin{bmatrix} 0 & 0 & 0 & \nabla_{u_4} g & \nabla_{u_5} g & \nabla_{u_6} g \end{bmatrix}$, for option 1 (C3.1).
- $\mathbf{H}^\top = \begin{bmatrix} 0 & 0 & 0 & 1 & \frac{\nabla_{u_5} g}{\nabla_{u_4} g} & \frac{\nabla_{u_6} g}{\nabla_{u_4} g} \end{bmatrix}$, for option 2 (C3.2).
- $\mathbf{H}^\top = \begin{bmatrix} 0 & 0 & 0 & \left| \frac{\partial \mathbf{u}}{\partial (GOR_3)} \right|^\top \end{bmatrix}$, for option 3 (C3.3).

No single input, $w_{gl,i}$, is assigned to each active constraint for direct constraint control. However, there are calculated inputs, i.e., $w_{gl,w_{pg},dir}$, which are used to adjust the last three inputs when the constraint g is active.

Master (*auxiliary*-) constraint controller

To determine $\boldsymbol{\lambda}$, we use a PI controller as a master (*auxiliary*-) constraint controller. We do not include `max` selector and anti windup as the disturbance scenario in this simulation focuses on active constraint handling.

Steady-state gradient estimation techniques

To estimate the required steady-state gradients, we use the model-based gradient estimation framework proposed in [55]. Note that the proposed framework is not restricted to this gradient estimation approach, so one may instead use any other model-based or model-free gradient estimation scheme [35]. We assume that disturbance and differential states are accurately estimated.

PID controllers setup

PID controllers are tuned using the SIMC tuning method introduced by [52]. The local gradient controllers, the master *auxiliary* controller, and the override/direct constraint controller are designed with a sampling time of 1 sec.

Simulation setup

The plant simulator is developed using the `CasADi ver.3.5.1` toolbox ([53]) in `MATLAB R2019b`, and is simulated using the `IDAS` integrator. The simulations are performed on a `2.11 GHz` processor with `16 GB` memory for 10 hours simulation time.

The system is subject to disturbances, i.e., gas oil ratio (GOR) for all wells are time-varying as shown in Figure 7.4. A dramatic disturbance occurs at $t = 2$ hr, where both GORs from well 1 and 5 increase 15%, GOR from well 2 increases 12%, and GOR from well 3 decreases 5%.

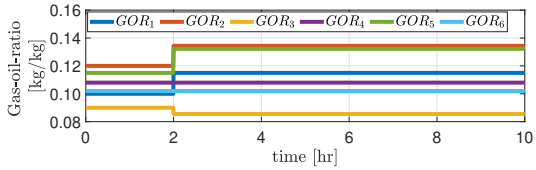


Figure 7.4: GOR variations (disturbances) in the six wells.

Simulation results

In Figure 7.5-7.7, we present simulation results comparing different control approaches: C1, C2, C3.1, C3.2, and C3.3. Despite all approaches starting and converging to the same steady-state optimal (i.e., constraint limits) after a disturbance originating from a GOR change at $t = 2$ hr, notable differences emerge.

Table 7.1: Constraint Satisfaction Performance for the first 10 hrs.

	C1	C2	C3.1	C3.2	C3.3
Max. Violation [kg/s]	0.7552	0.8849	0.7308	0.7343	0.7980
Int. Violation [kg]	190.47	273.27	166.91	170.08	241.21
Ave. Violation [kg]	0.0053	0.0076	0.0046	0.0047	0.0067
RMSE [kg]	0.0551	0.0637	0.0479	0.0485	0.0631

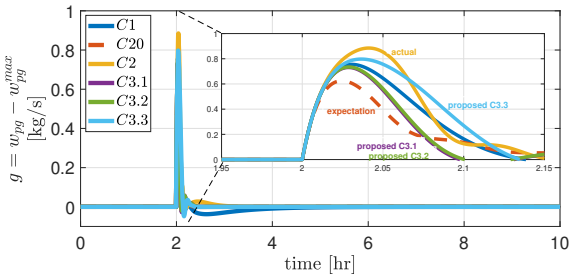


Figure 7.5: Constraint Satisfaction Performance

C2, a single-input approach, falls short in significantly reducing the maximum violation due to input rate constraints, as evident from the dashed yellow line compared to the red line at around $t = 2$ hr in Figure 7.6. On the other hand, multi-input options C3.1 and C3.2 exhibit comparable effectiveness in minimizing the maximum violation. However, C3.3 disappoints by failing to outperform the option without any override technique (C1), though it still fares better than C2. The subpar performance of C3.3 is attributed

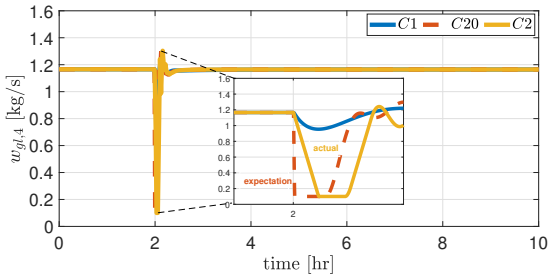


Figure 7.6: Overridden Input for Single-Input

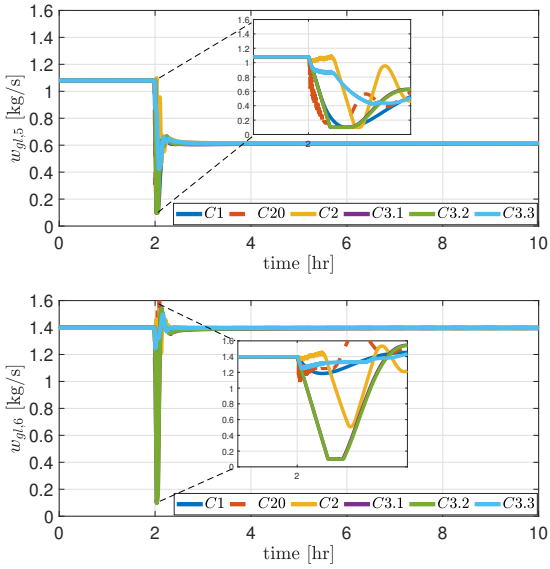


Figure 7.7: Input 5 and Input 6 employed as overridden inputs in multi-input approaches.

to insufficient interaction facilitated by the coupling constraint, leading to an inadequate representation of disturbances from other wells.

For instance, disturbances originating from other wells, excluding well 4, are not adequately captured by the disturbance originating from well 4. Numerically, the fourth element of matrix $\mathbf{H}$ is significantly larger than the fifth and sixth elements. Despite C3.3's limitations, overall, multi-input approaches are preferable compared to the single-input one due to their ability to jointly control constraints. This is evident in Figure 7.7, where inputs from both well 5 and 6 are used to minimize export gas overflow (constraint g).

Notably, the numerical analysis in Table 7.1 offers a more detailed comparative study. C3.1 and C3.2 stand out by reducing the maximum constraint violation by $2.77 - 3.23\%$, integrated constraint violation by $11.8 - 13.9\%$, and the average violation by $11.3 - 13.2\%$ over the first 10 hours. Focusing on active constraint handling during this period, both

approaches demonstrate a substantial reduction in root mean square error (RMSE) by $12.9 - 13.8\%$.

7.4.2 Single vs Multi-input in Active Constraint Switching

Problem and case description

This second numerical simulation includes and focuses active constraint switching. We consider the same case study as the one utilized in the previous subsection. However, in addition to constraint (7.12b) (linked to constraint function g_2), we also consider maximum available power for gas lift compressor, labelled as Pow_{gl}^{max}, as another coupling constraint (linked to constraint function g_1). As a result, this problem is encapsulated in the following optimization problem:

$$\min_{w_{gl,i},\, \forall i \in N} \quad J_N = -\sum_{i \in N} \$_{o,i} w_{po,i} + \sum_{i \in N} \$_{gl,i} w_{gl,i} \tag{7.13a}$$

$$\text{s.t.} \quad g_1(\mathbf{u}, \mathbf{d}) =: Pow_{gl} - Pow_{gl}^{max} \leq 0, \tag{7.13b}$$

$$g_2(\mathbf{u}, \mathbf{d}) =: \sum_{i \in N} w_{pg,i} - w_{pg}^{max} \leq 0 \tag{7.13c}$$

Similarly, a manipulated variables (MV) are prioritized to control critical constraints, namely, $w_{gl,5} \leftrightarrow g_2$, and $w_{gl,2} \leftrightarrow g_1$.

In addition to gas-oil ratio (GOR), we also consider that the maximum available power for the gas-lift compressor is varying with time due to changing power load balance in the processing facility. Furthermore, the total produced gas is limited by the maximum capacity handling of the gas processing facility. This limit is also assumed to be time-varying.

The GOR for all wells vary as shown in Figure 7.8, where it can be seen that the system is frequently subject to disturbances, and a dramatic drop occurs at $t = 6$ hr. The available power for the gas-lift compressor (Pow_{gl}^{max}) and gas processing capacity (w_{pg}^{max}) also varies, which affects the optimal allocation of the gas-lift.

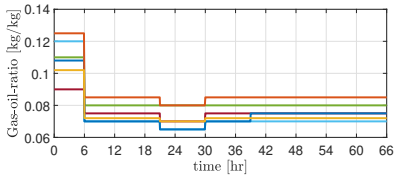


Figure 7.8: GOR variations (disturbances) in the six wells.

To maintain the focus of this work, we compare the performance of the following approaches:

- (C0) The ideal steady-state optimal. To obtain the ideal steady-state optimal solutions (C0), we solve problem (7.13) every 150 seconds.
- (C1) The standard dual-based *dFoS* (No direct/override constraint control). To solve problem (7.13) using C1, one can read [41].
- (C2) The dual-based *dFoS* equipped with single-input override constraint control.

- (C3.1) The dual-based *dFoS* equipped with multi-input override constraint control. It is not necessary to include C3.2 and C3.3 as C3.1 is comparable to C3.2, and C3.3 is not better than C3.1.

The last two approaches are explained the following sections.

Single-input override constraint control in the dual-based *dFoS* framework

C2 is a special case of the multi-input approach. We choose to assign $w_{gl,2}$ as the only input that is responsible for tightly controlling active constraint Pow_{gl} to Pow_{gl}^{max}, and $w_{gl,5}$ as the only input that is responsible for tightly controlling active constraint R_{pg_N} to w_{pg}^{max}. Defining g_1, and g_2 as the first and second row of $\mathbf{g}$, these pairings can also be described in matrix formulation, where matrix $\mathbf{H}$ is chosen as follows.

$$\mathbf{H}^\top = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}$$

Multi-input override constraint control in the dual-based *dFoS* framework

In multi-input approach (C3.1), we choose $\mathbf{H}^\top = \nabla_{\mathbf{u}}^\top \mathbf{g}_{\mathbb{A}}$. Thus, no single input, $w_{gl,i}$, is assigned to each active constraint for direct constraint control. However, there are calculated inputs, i.e., $w_{gl,pow_{gl},dir}$ and $w_{gl,w_{pg},dir}$, which are used to adjust all the inputs, labeled by $\mathbf{w}_{gl}$, if any of those constraints is active.

In this case, the two constraints (7.13b)-(7.13c) are never active simultaneously. Thus, we assign all available inputs to tightly control the active one. In this case, the selected correction factor is

$$\Delta \mathbf{w}_{gl} = \min\left(\Delta \mathbf{w}_{gl,pow_{gl}}, \Delta \mathbf{w}_{gl,w_{pg}}\right)$$

where $\Delta \mathbf{w}_{gl,pow_{gl}}$ and $\Delta \mathbf{w}_{gl,w_{pg}}$ are the correction factors when g_1 and g_2 are active, respectively. The selection matrices are $\mathbf{H} = \nabla_{\mathbf{w}_{gl}}^\top g_1$ to calculate $\Delta \mathbf{w}_{gl,pow_{gl}}$, and $\mathbf{H} = \nabla_{\mathbf{w}_{gl}}^\top g_2$ to calculate $\Delta \mathbf{w}_{gl,w_{pg}}$.

Master (*auxiliary*-) constraint controller

To determine $\boldsymbol{\lambda}$, we use a PI controller as a central constraint controller with a max selector that gives $\boldsymbol{\lambda} = \mathbf{0}$ when the constraint is no longer optimally active. The anti-windup is necessary to avoid $\boldsymbol{\lambda}$ keeps changing in this case. Thus, this selector gives a value of either 0 or $\hat{\boldsymbol{\lambda}}$.

$$\begin{aligned} \hat{\boldsymbol{\lambda}} &= \boldsymbol{\lambda}^k + \boldsymbol{K_P}\left(\mathbf{H}^{c\top}\mathbf{w}_{gl,ind} - \mathbf{w}_{gl,dir}\right)^k \\ &+ \sum_{\tau=k-1}^{k}\left(K_I\left(\mathbf{H}^{c\top}\mathbf{w}_{gl,ind} - \mathbf{w}_{gl,dir}\right)^\tau + K_{aw}\left(\boldsymbol{\lambda} - \hat{\boldsymbol{\lambda}}\right)^\tau\right) \end{aligned} \tag{7.14}$$

where k is the current step. K_P, K_I, and K_{aw} are proportional, integral and anti wind-up gain, respectively.

Simulation results

Figure 7.9 shows the simulation results comparing the ideal optimum, single-input approach (C2), and multi-input approach option 1 (C3.1), where we can notice that after a

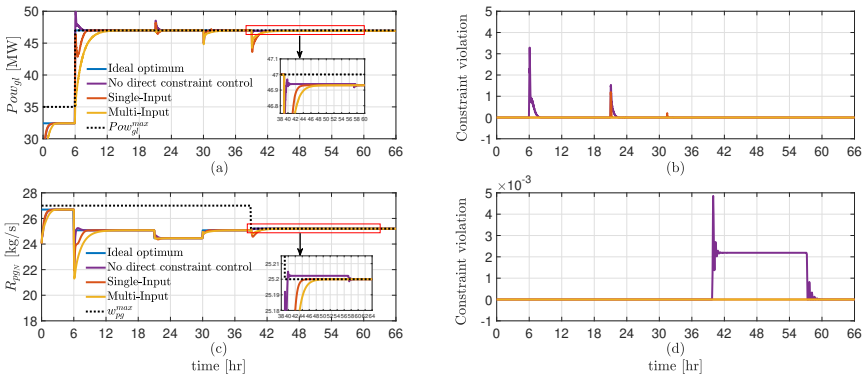


Figure 7.9: Simulation results showing the constraint tracking and control performance of the multi-input approach (C3.1) compared to single-input (C2), the standard dual *dFoS* framework (C1) and ideal optimum (C0), along with the error to the ideal one (right-hand plots)

dramatic GOR drop at $t = 6$ hr, more power is required by the gas-lift compressor to supply more lift gas. Thus, the maximum available power constraint is active. Note that the single-input approach does not work 'properly' to control the constraint during the transient due to input rate saturation. On the other hand, the multi-input approach (C3.1) can control the active constraint tightly. In addition, when the maximum gas handling capacity drops and GOR of well 1 increases at $t = 39$ hr, the active constraint region switches, and the maximum gas handling capacity is active. Without changing the control structure, the multi-input approach (C3.1) can automatically switch its 'mode'. After switching, this approach tightly controls the new active constraint. We can also notice that there is no significant dynamic violation in this case, so that the back-off parameters are relatively small, which means reducing more potential loss.

Figure 7.10 shows that probably the only issue of multi-input (in this case) is the overshoot of dual variables at $t = 6$ hr, that appears due to disturbance and the presence of multi-input direct/override constraint control itself. Regarding active constraint switching, Figure 7.10 (b) indicates that both single-input (C2) and multi-input (C3.1) can minimize dynamic violation at $t = 39$ hr with a relatively short transient time. Meanwhile, the standard dual *dFoS* framework (C1) consumes more transient time for switching.

Finally, Figure 7.11 compares and confirms that the multi-input is the approach that can address the issue of input rate saturation indicated by its capability to reach the optimal-steady state condition, and minimize the back-off.

Regarding economic loss due to back-off, one can notice in Figure 7.9(c) that the standard dual *dFoS* framework (C1) needs to apply around 3.2 MW of back-off of the real Pow_{gl}^{max}, single-input (C2) around 1.1 MW, and multi-input (C3.1) almost 0 MW in this case. This back-off has included the violation shown in Figure 7.9(d) because the change of the cost shown in Figure 7.11(a) of the second constraint is less than the first one. After implementing back-off strategy, both the standard dual *dFoS* framework (C1), and single-input (C2) lose around \$369/hr, and \$104/hr, respectively. Meanwhile, the multi-input approach (C3.1) loses almost \$0/hr in this case.

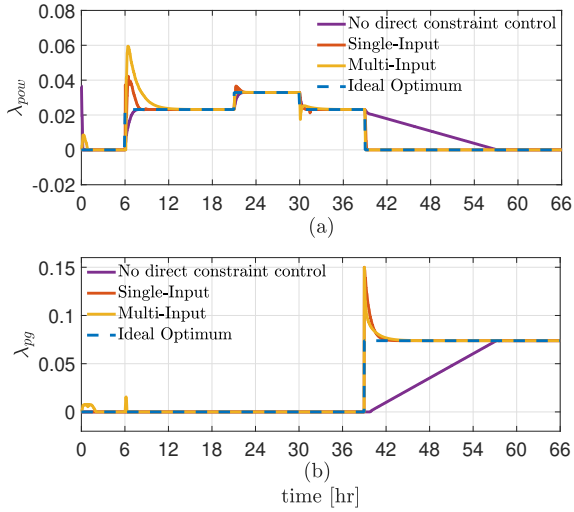


Figure 7.10: The comparison of the Lagrange multiplier of the single-input (C2), multi-input (C3.1), the standard dual *dFoS* framework (C1), and ideal optimum approaches (C0).

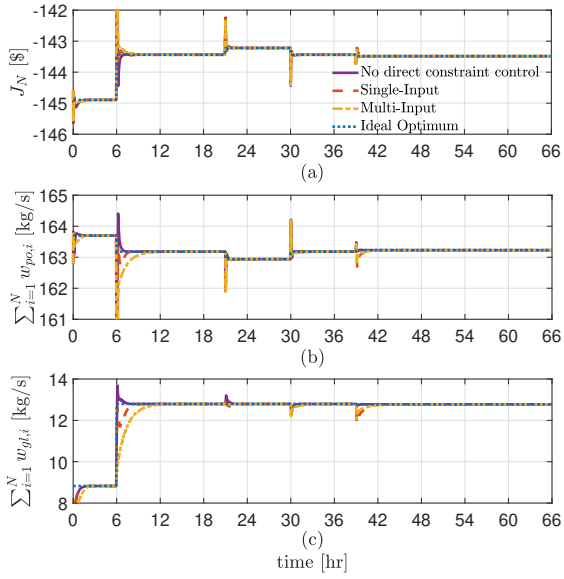


Figure 7.11: The overall performance of the multi-input approach (C3.1) compared to the single-input (C2), the standard dual *dFoS* framework (C1), and ideal optimum (C0).

7.5 Chapter Summary

In this paper, we proposed a multi-input direct/override constraint control that is combined with dual-based *dFoS* framework. We showed that the proposed control structure can overcome the issue of input rate saturation by introducing an online correction factor for the assigned inputs. Since the correction factor is constructed by a selection matrix, several inputs may jointly contribute to 'directly' controlling the active constraint, thus, avoiding the dependency on a single input. This strategy enables system-wide optimal operation with minimum back-off even under saturated inputs rate conditions, and without losing the flexibility of active constraint switching.

Chapter 8

Primal-based dFoS Framework with Compensator: Minimizing Dynamic Violation

In the realm of distributed feedback-optimizing system framework, this chapter suggests other possible approach to handle critical constraint which is based on primal decomposition equipped with compensator subsystem. This chapter is based on the work in [91].

8.1 Introduction

As shown previously, chapter 4-7 explore and exploit the potential of dual decomposition in the construction of distributed feedback-optimizing system framework. The aim of those chapters starting from transforming interactive to non-interactive problem, numerical-based optimization to feedback-based optimization, and ending with handling better constraints in a more realistic cases. As a continuation of our work, this chapter, chapter 9, and chapter 10 explore and exploit the potential of primal decomposition in the construction of distributed feedback-optimizing system framework. Specifically, the main contribution of this chapter is constructing what so called primal-based distributed feedback-optimizing system framework, that achieves optimal steady-state operation in a distributed manner, without the need to solve numerical optimization problems online and with minimum dynamic constraint violation.

8.2 Problem Formulation

In this section, we describe the optimization problem for the entire system consisting of a network of N subsystems. These subsystems are denoted by the set $\mathcal{N} = \{1, \ldots, N\}$. In practice, these subsystems are usually operated by different companies or organization and therefore, we assume each subsystem is optimized locally. For example, four of seven oil wells are operated by company A, and the remaining are operated by company B. Typically, each company has its own best practice in optimizing their wells (also known

as local optimization).

Let subsystem i be modeled as a nonlinear state-space system.

$$\begin{aligned} \dot{\mathbf{x}}_i &= \mathbf{f}_i(\mathbf{x}_i, \mathbf{u}_i, \mathbf{d}_i) \\ \mathbf{y}_i &= \mathbf{h}_i(\mathbf{x}_i, \mathbf{u}_i, \mathbf{d}_i) \end{aligned} \tag{8.1}$$

where $\mathbf{x}_i \in \mathbb{R}^{n_{x,i}}$, $\mathbf{u}_i \in \mathbb{R}^{n_{u,i}}$, $\mathbf{d}_i \in \mathbb{R}^{n_{d,i}}$ and $\mathbf{y}_i \in \mathbb{R}^{n_{y,i}}$ denote the vector of states, inputs, disturbances/parameters and available measurements of each subsystem, respectively. Each subsystem may also have local constraints.

We consider the overall network as a nonlinear state-space system and define all inputs, states, and disturbances as shown in the following.

$$\mathbf{u} = \left[\mathbf{u}_1, \ldots, \mathbf{u}_N\right]^\top; \mathbf{x} = \left[\mathbf{x}_1, \ldots, \mathbf{x}_N\right]^\top; \mathbf{d} = \left[\mathbf{d}_1, \ldots, \mathbf{d}_N\right]^\top \tag{8.2}$$

The steady-state optimization problem is

$$\min_{\mathbf{u}_i, \forall i \in \mathcal{N}} \quad J_{\mathcal{N}} = \sum_{i \in \mathcal{N}} J_{\mathcal{N}_i} \tag{8.3a}$$

$$\text{s.t.} \quad \mathbf{f}(\mathbf{x}, \mathbf{u}, \mathbf{d}) = 0, \tag{8.3b}$$

$$g(\mathbf{x}, \mathbf{u}, \mathbf{d}) \leq 0 \tag{8.3c}$$

where constraint (8.3b) is related to the entire system model, and constraint (8.3c) is a (coupling) (in-)equality constraint.

8.3 Proposed Solution

8.3.1 Distributed Optimization using Primal Decomposition

Solving the integrated optimization problem (8.3) requires a detailed model and their interactions in addition to the constraints and measurements, which may be undesirable or unnecessary in the practical context. Therefore, we propose to solve problem (8.3) in a distributed manner by decomposing the problem. In this paper, we propose an online optimization method, i.e., using simple feedback controllers, based on primal decomposition and addressing the issue of primal infeasibility.

First, we introduce a virtual subsystem denoted as subsystem 0, in which the cost function is $J_{\mathcal{N},0} = 0$. As a consequence, we define the set $\mathcal{N}_0 = \{0, 1, \ldots, N\}$.

Defining constraint (8.3c) as linear constraint (if it is non-linear, one can consider to linearize it at the operating point), $g(\mathbf{x}, \mathbf{u}, \mathbf{d}) = \sum_{i=1}^{N} g_i(\mathbf{x}_i, \mathbf{u}_i, \mathbf{d}_i) - g^{max}$ (additive w.r.t the contributions of the subsystems), we introduce a slack variable, $\mathbf{g}_0$, to convert any inequality constraint in (8.3c) into equality constraints, where $g(\mathbf{x}, \mathbf{u}, \mathbf{d}) + g_0 = 0$. This modification does not change the structure that (8.3a) is additively separable in the cost, and the system model (8.3b) are imposed for each subsystem independently.

By providing an initial value of local constraint for the variables of the coupling constraint, labeled by g_i^{sp}, where $g^{sp} = \sum_{i=1}^{N} g_i^{sp}$, and letting a central problem deal with the active coupling constraint satisfaction, integrated optimization problem (8.3) can be seen

as the following separable problem.

$$\min_{\mathbf{u}_i, \forall i \in \mathcal{N}_0} \quad J_{\mathcal{N}} = \sum_{i \in \mathcal{N}} J_{\mathcal{N},i} \tag{8.4a}$$

$$\text{s.t.} \quad \mathbf{f}_i(\mathbf{x}_i, \mathbf{u}_i, \mathbf{d}_i) = 0, \forall i \in \mathcal{N}, \tag{8.4b}$$

$$g_i(\mathbf{x}_i, \mathbf{u}_i, \mathbf{d}_i) - g_i^{sp} = 0, \forall i \in \mathcal{N}_0, \tag{8.4c}$$

$$\sum_{i \in \mathcal{N}_0} g_i^{sp} = g^{max} \tag{8.4d}$$

Note that here we introduce auxiliary primal variables g_i^{sp}. Moreover, as long as Eq. (8.4d) is satisfied, the primal feasibility of the coupling constraint (8.3c) is guaranteed.

By relaxing the local constraint (8.4c), problem (8.4) can be re-written as a Lagrange function that can be decomposed into smaller subproblems, and each subproblem solves the optimization problem for subsystem i.

$$\mathcal{P}_i(g_i^{sp}) := \min_{\mathbf{u}_i} \mathcal{L}_i(\mathbf{u}_i, g_i^{sp}, \lambda_i) \tag{8.5}$$

where $\mathcal{L}_i(\mathbf{u}_i, g_i^{sp}, \lambda_i) = J_{\mathcal{N},i} + \lambda_i g_i(\mathbf{x}_i, \mathbf{u}_i, \mathbf{d}_i)$. The local Lagrange multiplier, labeled by λ_i, is associated with local constraint (8.4c). The local constraint converges to the same value in steady-state optimal conditions.

8.3.2 Controllers and Estimators

Each subsystem solves its local optimization problem by considering the setpoints (auxiliary primal variables, g_i^{sp}) provided by the master/central constraint controllers.

Master/central constraint controllers

These controllers update the setpoints iteratively, based on given local Lagrange multipliers computed by each subproblem. The goal of these controllers in a central problem is to provide setpoints that satisfy the primal feasibility (8.4d).

$$\min_{g_0^{sp}, g_1^{sp}, \ldots, g_N^{sp}} \quad \sum_{i \in \mathcal{N}_0} \mathcal{P}_i(g_i^{sp}) \tag{8.6a}$$

$$\text{s.t.} \quad \sum_{i \in \mathcal{N}_0} g_i^{sp} = g^{max}, \tag{8.6b}$$

$$g_0^{sp} \geq 0 \tag{8.6c}$$

where $\mathcal{P}_i(g_i^{sp})$ is given by (8.5), constraint (8.6b) comes from Eq.(8.4d), and constraint (8.6c) forms the foundation for using the `max` selector in the subsequent explanation of the virtual subsystem.

Compensator subsystem:

To ensure primal feasibility, one local setpoint (e.g., subsystem N) is given by

$$g_N^{sp,k+1} = g^{max} - \left(g_0^{sp,k+1} + \ldots + g_{N-1}^{sp,k+1} \right) \tag{8.7}$$

We call this subsystem as compensator subsystem.

Normal subsystem:

Each local setpoint g_i^{sp} at time step $k+1$ can be determined using the steepest descent direction of the central problem (8.6a), which is given by the subgradient. For $j = \{0, ..., N-1\}$,

$$\nabla_{g_j^{sp}} \left(\sum_{i \in \mathcal{N}_0} \mathcal{P}_i(g_i^{sp,k}) \right) = -\lambda_j^k + \lambda_N^k, \tag{8.8}$$

The updated local setpoint at the next time step is,

$$g_i^{sp,k+1} = g_i^{sp,k} + K_{I,i} \nabla_{g_i^{sp}} \left(\sum_{i \in \mathcal{N}_0} \mathcal{P}_i(g_i^{sp,k}) \right) \tag{8.9}$$

where we may consider an integrating controllers with integral gain $K_{I,i} = \frac{1}{K_i(\tau_{c,i})}$, and K_i is the step response gain, and $\tau_{c,i}$ is the desired closed-loop time constant. Note that the desired time constant should be slow enough to satisfy the time-scale separation concept [85]. This concept is necessary to avoid undesired behaviors such as oscillatory and deviating behavior.

Note that to compensate any change in the normal subsystem, we assume that each subsystem informs its local Lagrange multipliers λ_i^k to the compensator subsystem, and receive the local Lagrange multipliers λ_N^k of the compensator subsystem.

Virtual subsystem:

Since we introduce a slack variable g_0^{sp} to store un-utilized resource, and the storage is physically never been negative, it is necessary to use `max` selector as follows.

$$g_0^{sp,k+1} = \max \left[0, g_0^{sp,k} + K_{I,0} \nabla_{g_0^{sp}} \left(\sum_{i \in \mathcal{N}_0} \mathcal{P}_i(g_i^{sp,k}) \right) \right] \tag{8.10}$$

By implementing these strategies, i.e., compensator, normal, and virtual subsystem, the setpoints, provided by these controllers, guarantee the primal feasibility.

Local setpoint controllers

Given the local setpoint g_i^{sp}, the local setpoint controller regulates the actual local primal variables g_i to g_i^{sp}. The updated local input at the next time step $\mathbf{u}_i^{sp,k+1}$ is given by

$$\mathbf{u}_i^{sp,k+1} = \mathbf{u}_i^{sp,k} + K_{IL,i} \left(g_i - g_i^{sp,k} \right) \tag{8.11}$$

where we may consider an integrating controllers with integral gain $K_{IL,i} = \frac{1}{K_{L,i}(\tau_{cL,i})}$, and $K_{L,i}$ is the step response gain, and $\tau_{cL,i}$ is the desired closed-loop time constant. Typically, the desired time constant is designed as fast as possible. However, it is necessary to carefully choose the desired time constant $\tau_{cL,i}$ to ensure that the local setpoint controller does not too aggressively track the setpoint given by central constraint controllers.

Remark 8.1: Special Case

Note that the setpoint controller is not necessary in the special case, when we have a shared input constraint because the central constraint controller has provided the

optimal input.

Local Lagrange Multiplier estimation

Note that (8.8) requires local Lagrange multiplier estimates. In the traditional RTO framework, this is available when solving the numerical optimization problem. However, when using feedback control, this is not directly available, and hence must be estimated, which will be describe below. According to KKT (Karush-Kuhn-Tucker) conditions, the stationary point is reached when

$$\nabla_{\mathbf{u}_i} \mathcal{L}_i(\mathbf{u}_i, g_i^{sp}, \lambda_i) = 0$$

for all subsystems, and all λ_i converge to the same optimal value. Thus, the local Lagrange Multiplier λ_i can be computed as follows.

$$\lambda_i = -\nabla_{\mathbf{u}_i} J_{\mathcal{N},i} \left(\nabla_{\mathbf{u}_i} g_i(\mathbf{x}_i, \mathbf{u}_i, \mathbf{d}_i)\right)^{-1} \tag{8.12}$$

where the number of local manipulated variables must be equal to or more than the number of constraints in common, and the solution must be unique.

In order to evaluate (8.12), each subsystem i is required to estimate its local steady-state cost and constraint gradient, which can be achieved locally using any model-based or model-free gradient estimation. This estimation takes into account the effect of the updated input calculated in (8.11). For a list of gradient estimation techniques for RTO see [35], and [74].

Remark 8.2: Virtual Local Lagrange Multipliers

Virtual subsystem, λ_0 is always 0 because $J_{\mathcal{N},0}$ is defined as 0, and to limit the dual variable to be non-negative in a steady-state condition (i.e., to satisfy steady-state dual feasibility).

8.3.3 Online Primal Decomposition Framework

By combining the concept of primal decomposition, the idea of master/central constraint controllers, local setpoint controllers, and local Lagrange multiplier estimation as described above, we propose to solve the problem of real-time resource allocation in handling coupling constraint using distributed feedback-optimizing control using Primal decomposition framework. This framework theoretically can reach steady-state optimal condition and guarantees primal feasibility.

Fig. 8.1 illustrates the implementation of this framework in solving the above problem. The master/central constraint controllers, containing virtual, normal and compensator subsystems, provide new set points for local coupling constraint, g_i (see eq. (8.7),(8.9), and (8.10)). These set points will be tracked by local setpoint controllers (see eq. (8.11)). Should there be any disturbance $\mathbf{d}_i$, one can use the current plant information to estimate the plant's current state and parameters/disturbance using local dynamic estimator such as Extended Kalman Filter (EKF). Using the inputs, estimated states and parameters/disturbance, one can estimate both cost and constraint gradient to compute the local Lagrange multipliers as shown in eq. (8.12). Thereafter, these multipliers are used by master/central constraint controllers to determine the new setpoints.

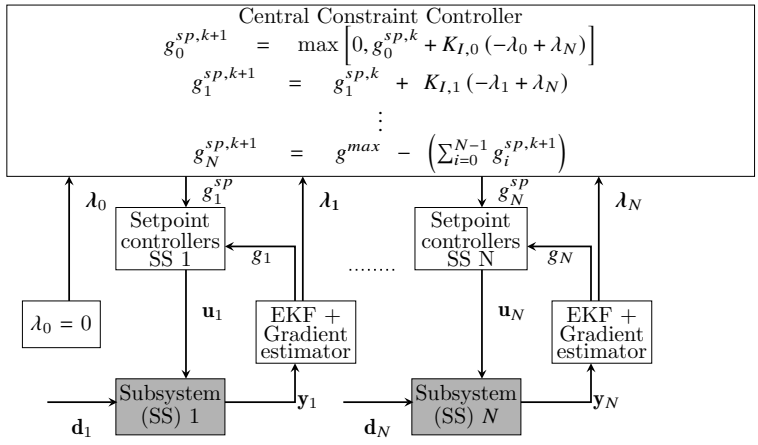


Figure 8.1: The proposed primal-based distributed feedback-optimizing system framework using simple feedback controllers and selector. The grey and white boxes represent the physical system and the computation block, respectively.

Remark 8.3: Types of shared resources

Note that we have two types of shared resources here, the shared input, that we need to optimally allocate, and the shared constraint, g, that all subsystem should cooperate optimally to satisfy. The shared input can be total flow of materials, steam, or energy that any kind of process industries usually need. The shared constraint can be any type of constraint that two or more subsystems have influence on, i.e., plant capacity. Specifically for the solid mining industry, we could consider the extracted earth deposit as the flow of materials, and the maximum capacity of the processing plant, e.g., smelter, as the shared constraint.

8.4 Numerical Example

We demonstrate the proposed method in a network of gas-lifted oil wells production system. These oil wells are operated locally and share a common processing facility at the topside. Using recent technology, the subsea production wells have capabilities for the measurement of the multi-phase flow rates (i.e. multiphase flowmeter, MPFM or virtual flowmeter, VFM technology solutions) at respective wellheads [92]. Since the export gas handling capacity available on a platform is usually limited, it is necessary to optimally allocate the lift gas among the different wells.

Note that the term of export and produced gas is interchangeable in this numerical example. Both terms represent the total of the gas produced by reservoir and the injected gas-lift.

In this section, we apply the proposed method control structure on a gas-lifted well network (liquid and gas extraction activities) with $N = 2$ wells, that are operated locally. The optimization objective of this case is to maximize total oil production, $w_{to} = \sum_{i=1}^{N} w_{po,i}$

while minimizing the cost of total gas lift, $w_{gl} = \sum_{i=1}^{N} w_{gl,i}$. Thus, $J_{\mathcal{N}_i} = -p_o w_{po,i} + p_{gl} w_{gl,i}$, where p_o, and p_{gl} are the oil price and the gas lift cost, respectively. The coupling constraint is $g(\mathbf{x}, \mathbf{u}, \mathbf{d}) = \sum_{i=1}^{N} w_{pg,i} - w_{pg}^{max}$, where $w_{pg,i}$ is the local export gas, and w_{pg}^{max} is the maximum capacity to handle total export gas. Fig. 8.2 illustrates this case study completed with the proposed control structure, where well 2 is assigned to ensure the setpoint primal feasibility (see Eq. (8.7)). Note that each well has an MPFM to measure the actual local export gas.

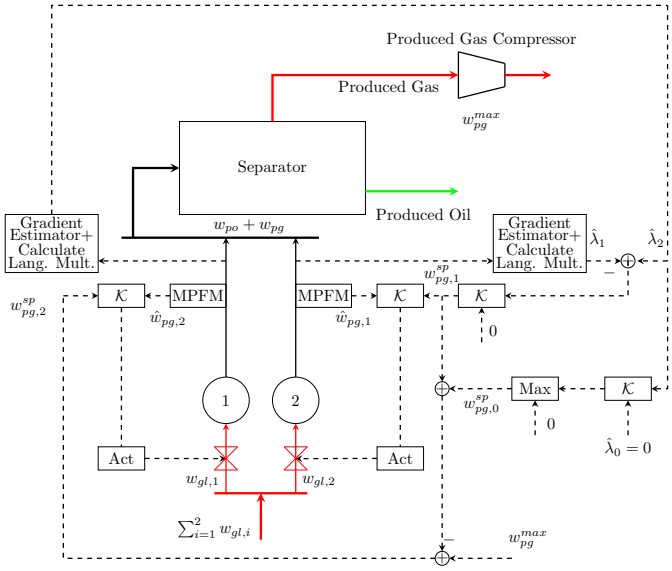


Figure 8.2: A simplified process diagram of a gas-lifted oil production network with constraints in maximum export gas handling capacity, and equipped with the proposed control structure. Dashed lines represent data transmission in the proposed control structure. Act stands for actuator. $\mathcal{K}$ represents controllers.

The gas-oil ratio (GOR), one of the essential reservoir properties, is a time-varying disturbance for the different wells (feed disturbance). Usually, the separator pressure is controlled. To accommodate this scenario, we have to consider optimization problem reformulation as described in Chapter 4. However, for simplification, we assume that the manifold pressure is controlled, and the setpoint is time-varying due to the change in the processing facility or other connected wells. Thus, we consider disturbance $\mathbf{d}$ consisting of GOR and manifold pressure, as shown in Figure 8.3. In addition to these disturbances, the produced gas processing capacity (w_{pg}^{max}) varies, which affects the optimal allocation of the lift gas. The disturbances may also lead to an unconstrained case, where the coupling constraint (8.3c) is inactive.

The controllers are tuned using SIMC (Simple/Skogestad Internal Model Control) rules introduced by [52]. The desired timescale of the local setpoint controllers is $\tau_{cL,1} = \tau_{cL,2} = 75$ seconds. To satisfy the timescale separation concept, the chosen timescale for the central constraint controllers is $\tau_{c,1} = \tau_{c,2} = 131.25$ seconds. Thus, the timescale ratio $\epsilon = \tau_{cL,i}/\tau_{c,i}$ is 0.5714. Since the time delay is insignificant, one could consider

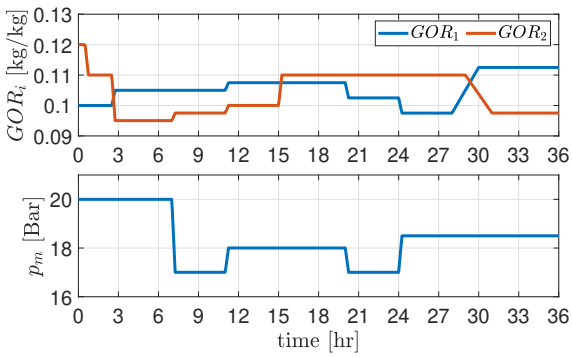


Figure 8.3: Disturbance.

integrating controllers. However, we use PI controllers in this simulation. Tab 8.1 displays the controllers' parameters.

Table 8.1: The Parameters of Controllers

Gain		Time Constant [sec]		Time Delay [sec]	
$K_{I,0}$	0.375	$\tau_{1,0}$	617	θ_0	0
$K_{I,1}$	0.375	$\tau_{1,1}$	617	θ_1	0
$K_{IL,1}$	1.031	$\tau_{1L,1}$	643	$\theta_{L,1}$	9
$K_{IL,2}$	1.039	$\tau_{1L,2}$	614	$\theta_{L,2}$	2

To estimate the local Lagrange multiplier, we execute three steps. First, we use the current plant information to estimate the plant's current state and parameters using EKF. Next, we use the updated model to evaluate the steady-state gradients. These first two steps utilize the same methods we use in [41]. Finally, we evaluate Eq. (8.12).

First, we solve the integrated production optimization problem (8.3) to obtain the ideal steady-state optimal setpoint as the baseline. Then, we implement the proposed framework described in Section 8.3.

Figure 8.4 shows the simulation results of the produced gas setpoints. These are the output of the central constraint controllers' performance, where we can observe that the total setpoint of the produced gas is not violating the constraint. As a consequence, the compensator subsystem (subsystem 2) 'absorbs' the violation, indicated by oscillations during transient. These associated oscillations can also be observed in Figure 8.5-8.6. Moreover, the produced gas setpoint of each well reaches the steady-state optimal setpoint labeled by $w_{pg,i}^{sp,\star}$. Furthermore, the steady-state slack variable $w_{pg,0}^{sp}$ also reaches 0 in the constrained case and $w_{pg,0}^{sp} > 0$ in unconstrained case.

Figure 8.5 depicts the simulation results showing the performance of the local setpoint controllers and the local Lagrange multiplier estimator. The top plot shows that the local setpoint controllers have successfully tracked the produced gas setpoints given by the central constraint controllers. The middle one shows that the manipulated variable, i.e., gas-lift

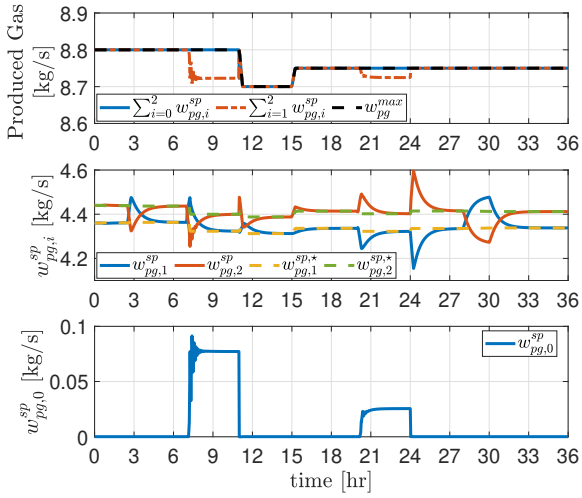


Figure 8.4: Top: Total produced gas optimal setpoint and its constraint. Middle: Produced gas setpoints. Bottom: Optimal unutilized produced gas capacity setpoint.

rates, reached the optimal steady-state conditions. Additionally, the local Lagrange multipliers converge to the optimal steady-state conditions and satisfy the dual feasibility in the steady-state shown in the bottom plot. Note that, in unconstrained case, the steady-state local Lagrange multiplier is 0. These results confirm the applicability of using a virtual variable to store the unutilized produced gas handling capacity.

Figure 8.6 displays the simulation results for the actual cost and the produced gas. The top plot shows that the presented method can reach the optimal steady-state cost. Moreover, the total produced gas satisfies the constraint with relatively short duration and insignificant magnitude violations during the transients. As mentioned above, this violation only depends on the tuning parameter we choose in the local setpoint controllers because the central constraint controllers have given the setpoints that guarantee the primal feasibility (see Fig. 8.4).

Fig. 8.7 shows the comparison with dual decomposition used in [41], where the central constraint controllers has to be slower (in timescale) than the presented method in active constraint switching (i.e., unconstrained to a constrained case). This requirement may lead to dynamic constraint violation when local gradient controllers are too aggressive, whereas central constraint controllers of the dual approach has no specific strategy to regulate the primal feasibility. During the transient, the Lagrange multiplier is suboptimal. This condition significantly contributes to constraint violation. Unlike dual, the central constraint controller of the presented approach ensures the total setpoint to satisfy the constraint. Thus, the 'small' violation is purely the product of the aggressive local setpoint controllers, which can be tuned *more independently*. Even this method does not need local setpoint controllers when it only has input constraints. Meanwhile, the dual approach may have an issue in tight constraint control as the central constraint controller has to be in a slow timescale. Forcing a faster timescale central constraint controller (larger ϵ) may lead

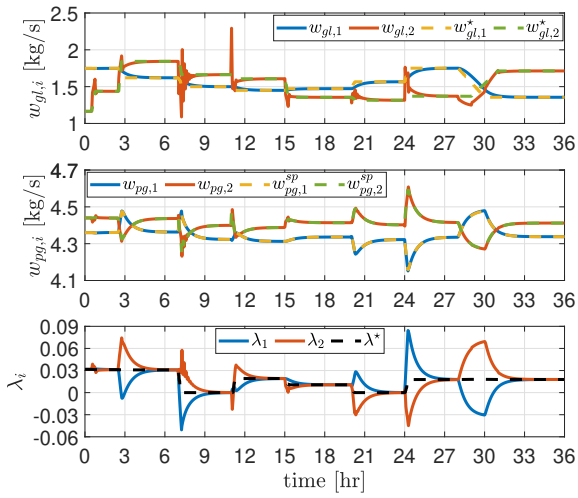


Figure 8.5: Top plot: Gas-lift rates ($u_i = w_{gl,i}$) and the optimal steady-state conditions. Middle plot: Produced gas rates and the setpoints. Bottom plot: The local Lagrange multipliers and the optimal steady-state.

to oscillatory behavior.

When it comes to solid extraction activities, one may consider a network of mines and smelters in a metal mining industry [93], which fundamentally has similar class of problem as the gas-lift well network. These mines produce concentrates that should be sold and transported to the smelters for processing. The optimization problem is to optimally allocate the raw materials from the mines in order to achieve maximum revenue since production capacity of the smelters is limited.

8.5 Chapter Summary

In this paper, we presented a primal-based *dFoS* framework. We showed that such a problem turns into a feedback control problem by introducing virtual subsystems or slack variables to store unutilized resources, implementing master/central constraint controls and local setpoint controls, and estimating Lagrange multipliers. The goals of central constraint controls are to directly control the constraint, update the local constrained variables setpoints, and regulate the primal feasibility of the constrained variables. The objective of local setpoint controls is to control constrained variables to the given setpoint. For the case study we consider in the simulation example, this proposed framework leads to a system-wide optimal operation without a numerical solver. Moreover, the setpoints provided by the master/central constraint controls satisfy the primal feasibility.

Assessment Table:

Let S-XI be the proposed framework of this chapter, the parameter we assess and compare are as described in section 2.5. Table 8.2 summarizes the assessment, where the approaches are:

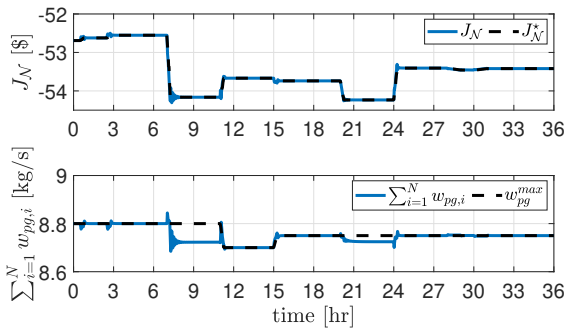


Figure 8.6: Top: Cost and the optimal steady-state conditions. Bottom: Actual total produced gas and its constraint.

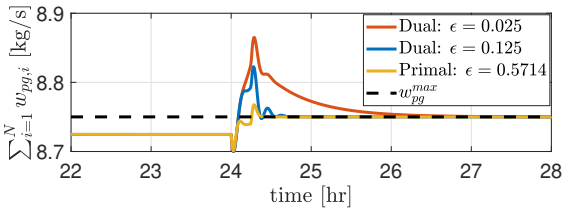


Figure 8.7: Comparison with different timescales separation (ϵ) and method. Small ϵ means the method of interest has slow timescale central constraint controller, and vice versa.

- S-IX: Dual-based Distributed Feedback-Optimizing System
- S-X: Dual-based Distributed Feedback-Optimizing System with Override
- S-XI: Primal-based Distributed Feedback-Optimizing System with Compensator

and the parameters we assess and compare are:

- (a): Is steady-state optimal achieved?
- (b): Is constraint controlled *transparently* (see Definition 2.1)?
- (c): Is flexible for active constraint switching?
- (d): Is constraint controlled *directly* (on fast time scale)?
- (e): Is applicable for less than twice differentiable Lagrange function?
- (f): Is numerical solver avoidable (computation time)?
- (g): Is it recommended for complex and large system?
- (h): Is input filter (or additional setpoint controller) avoidable?
- (i): Is parameters and states dynamic estimator avoidable?
- (j): Is gradient estimator avoidable?

Primal-based *dFoS* with compensator (S-XI) vs Dual-based *dFoS* with override (S-X):

From Table 8.2, we may conclude that both Primal-based *dFoS* with compensator (S-XI) and Dual-based *dFoS* with override (S-X) may produce comparable or equal performance. In fact, they are not. For instance, Primal-based *dFoS* with compensator is better

Table 8.2: Summary of Chapter 8

	S-IX	S-X	S-XI
(a)	Yes	Yes	Yes
(b)	Yes	Yes	Yes
(c)	Yes	Yes	Yes
(d)	No[1]	Yes	Yes
(e)	Yes	Yes	Yes
(f)	Yes	Yes	Yes
(g)	Yes	Yes	Yes
(h)	Yes	Yes	Yes
(i)	No	No	No
(j)	No	No	No

"Yes": the approach has satisfied the success parameter
"Yes": the approach has satisfied the success parameter in this Chapter
"No": the approach does not satisfied the success parameter
"No": does not satisfied the success parameter, and will be addressed in this thesis.
[1]: is also addressed in Chapter 3 with different approach.

when handling optimization problem with a shared input constraint, i.e, optimal resource allocation. It might also better when handling optimization problem with multiple shared input constraints as long as the each resources provision is affected by the other resource. However, Dual-based *dFoS* with override is more flexible in any type of optimization problem. For instance, for complex and interactive systems, Primal-based *dFoS* with compensator may fail if the matrix operation to determine the local Lagrange multiplier does not find unique solution.

Future works: As future works, Chapter 9 will not only show the experimental validation of primal-based *dFoS* equipped with compensator, but also compare it to the dual-based *dFoS* with and without override. Furthermore, an intriguing avenue of research involves the development of a *decentralized* feedback-optimizing system, where there is no single master constraint controller play as the central role. All distributed structures developed in Chapter 4, 6, 7, and 8 require a single central controller. Should there any information in the master level is not given to this controller, optimal operation will not be achieved. This study will be the focus of Chapter 10.

Chapter 9

A Comparative Study of DFoS Frameworks: An Experimental Validation in an Oil Well Rig

The goal of this chapter is to experimentally validate and compare the approaches described in Chapter 4, 6, and 8 This chapter is based on the works in [94, 95]

9.1 Introduction

The distributed feedback-optimizing control scheme holds several crucial benefits for process control (in order of the importance):

1. This scheme enables formulating each local problem independently (including independent local estimator), assuming that the overall objective consists of local objectives that are additively separable [13].
2. Fast local responses are achievable because one may use different closed-loop time constants for each local system.
3. Solutions can be implemented immediately after each iteration, eliminating the need to wait for convergence over several iterations.
4. This scheme circumvents slow response and numerical issues like divergence, often encountered in centralized Real-Time Optimization (RTO) methods (for example, the solution is not valid anymore due to fast disturbance).
5. The distributed feedback-optimizing control scheme may serve as an alternative to numerical-based RTO approaches as it can be computationally expensive in certain case.

Building on our previous work in Chapter 4, 6, and 8, the main contribution of this work is to experimentally validate and compare the three schemes of distributed feedback-optimizing control on a lab-scale gas-lifted oil well rig consisting of three wells with the lift gas being the shared resource that couples the three wells together.

9.2 Overview of Frameworks

Consider an integrated steady-state optimization problem of N different subproblems (subsystem).

$$\min_{\mathbf{u}} \quad J(\mathbf{u}, \mathbf{d}) = \sum_{i=1}^{N} J_i(u_i, d_i) \tag{9.1a}$$

$$\text{s.t.} \quad \mathbf{g}(\mathbf{u}, \mathbf{d}) = \sum_{i=1}^{N} \mathbf{g}_i(u_i, d_i) - \bar{\mathbf{g}} \leq 0 \tag{9.1b}$$

where $\mathbf{u} \in \mathbb{R}^{n_\mathbf{u}}$ are the set of manipulated variables, $\mathbf{d} \in \mathbb{R}^{n_\mathbf{d}}$ denotes the set of disturbances, $J : \mathbb{R}^{n_\mathbf{u}} \times \mathbb{R}^{n_\mathbf{d}} \to \mathbb{R}$ is the cost function, $\mathbf{g} : \mathbb{R}^{n_\mathbf{u}} \times \mathbb{R}^{n_\mathbf{d}} \to \mathbb{R}^{n_\mathbf{g}}$ denotes the (coupling) constraints, and $\bar{\mathbf{g}} \in \mathbb{R}^{n_\mathbf{g}}$ is the conflicting variables or the limit of the constraints.

Defining $\mathbf{u} = \begin{bmatrix} u_1, \ldots, u_N \end{bmatrix}^\top$ and $\mathbf{d} = \begin{bmatrix} d_1, \ldots, d_N \end{bmatrix}^\top$, $u_i \in \mathbb{R}^{n_{u_i}}$ denotes the decision variables and n_{u_i} is the number of decision variables in subsystem i, and $d_i \in \mathbb{R}^{n_{d_i}}$ denotes the disturbances and n_{d_i} is the number of the disturbances in subsystem i.

Further, $J_i : \mathbb{R}^{n_{u_i}} \times \mathbb{R}^{n_{d_i}} \to \mathbb{R}$ is a function that denotes the local objective of subsystem i, $g_i : \mathbb{R}^{n_{u_i}} \times \mathbb{R}^{n_{d_i}} \to \mathbb{R}^{n_\mathbf{g}}$ is a function that denotes the (in-)equality constraints in subsystem i, $n_\mathbf{g}$ is the number of the coupling constraints.

This works aims to translate a large-scale optimization problem (9.1) into simple feedback control problems, that can be solved using a simple tool(s), i.e., PID controllers and selector, or a configuration of them. To *distributedly* solve problem (9.1) using feedback control schemes, three structures of distributed feedback-optimizing control are available as described in the following sections.

9.2.1 Primal-based Distributed Feedback-optimizing System

This structure is based on the method proposed and discussed in [84, 91]. To be able to implement primal-based distributed feedback-optimizing control, the subsystem should satisfy Assumption 9.1.

Assumption 9.1: Sufficient local MVs

Each local system (subproblem) has enough input/manipulated variables (decision variables), u_i, to control the active coupling constraints, $\mathbf{g}$. Mathematically, $n_{u_i} \geq n_\mathbf{g}$, $i = 1, \ldots, N$.

By introducing a virtual subsystem (subsystem 0), one can reformulate problem (4.1) as an equality constraint problem as follows.

$$\min_{u_1, \ldots, u_N} \quad J_0 + \sum_{i=1}^{N} J_i(u_i, d_i) \tag{9.2a}$$

$$\text{s.t.} \quad \mathbf{g}_0 + \sum_{i=1}^{N} \mathbf{g}_i(u_i, d_i) - \bar{\mathbf{g}} = 0, \tag{9.2b}$$

$$\mathbf{g}_0 \geq 0 \tag{9.2c}$$

where J_0 is a constant and therefore will not influence any optimal solution, and $\mathbf{g}_0$ acts as a storage of unused resources/values (which is non-negative) that transform the inequality into an equality constraints (slack variables).

Subproblems as feedback control problems: Utilizing primal decomposition method, one can decompose problem (9.2) into N subproblem (local system), and each local system i has the following local subproblem.

$$\min_{u_i} \quad J_i\left(u_i, d_i\right), \qquad i = 1, \ldots, N \tag{9.3a}$$

$$\text{s.t.} \quad \mathbf{g}_i\left(u_i, d_i\right) = \mathbf{t}_i, \quad i = 1, \ldots, N \tag{9.3b}$$

Each subsystem i solves problem (9.3) using feedback-optimizing control, i.e., reduced gradient control, by executing the following steps.

1. Control $\mathbf{g}_i(u_i, d_i)$ to a setpoint of $\mathbf{t}_i$ using $n_{\mathbf{g}}$ degree of freedom.
2. For remaining $n_{u_i} - n_{\mathbf{g}}$ degree of freedoms, control reduced gradient, $\mathbf{N}_i^\top \nabla_{u_i} J_i\left(u_i, d_i\right)$, to setpoint of $\mathbf{0}$, where $\mathbf{N}_i^\top$ is the null space of $\nabla_{u_i} J_i\left(u_i, d_i\right)$ [23, 96].

Remark 9.1: Pairing

The pairing can generally be done in practice by using some rule of thumb, i.e., pair-close rule, non-negative relative gain array (RGA), and pair on large gain [21].

Remark 9.2: Special case

In the special case of $n_{u_i} = n_{\mathbf{g}}$, reduced gradient controller is not required.

Computing Lagrange multipliers: Updating the setpoint $\mathbf{t}_i$, in each subsystem requires the Lagrange multipliers corresponding to (9.3b). To compute Lagrange multipliers in a feedback control setting, the following conditions are required.

Assumption 9.2: Clear time scale separation

There is a clear time scale separation between the local controllers in each subsystem, and the controllers in the layer above used for coordination.

Note that the above assumption can be satisfied by tuning the local controllers 5 - 10 times faster than the controllers in the coordination layer [46, 85, 65, 57].

Assumption 9.3: The steady-state constraint gradient

The steady-state constraint gradient, $\nabla_{u_i} \mathbf{g}_i\left(u_i, d_i\right)$, is non-singular matrix.

If Assumption 9.2 hold, the stationary condition is achieved.

The Lagrangian of problem (9.3) is as follows.

$$\mathcal{L}_i(u_i, d_i, \mathbf{t}_i, \boldsymbol{\lambda}_i) = J_i\left(u_i, d_i\right) + \boldsymbol{\lambda}_i^\top \left(\mathbf{g}_i\left(u_i, d_i\right) - \mathbf{t}_i\right) \tag{9.4}$$

where $\boldsymbol{\lambda}_i \in \mathbb{R}^{n_{\mathbf{g}}}$ is a vector of Lagrange multipliers of the constraints.

At stationary condition $\nabla_{u_i}\mathcal{L}_i(u_i, d_i, \mathbf{t}_i, \boldsymbol{\lambda}_i) = \mathbf{0}$. Thus,

$$\nabla_{u_i} J_i\left(u_i, d_i\right) + \boldsymbol{\lambda}_i^{\top} \nabla_{u_i} \mathbf{g}_i\left(u_i, d_i\right) = 0 \tag{9.5}$$

If Assumption 9.3 hold, one can compute the local Lagrange multipliers as follows.

$$\boldsymbol{\lambda}_i^{\top} = -\nabla_{u_i} J_i\left(u_i, d_i\right)\left[\nabla_{u_i} \mathbf{g}_i\left(u_i, d_i\right)\right]^{-1} \tag{9.6}$$

Note that each subsystem estimates the cost and constraint gradients locally using only the local model and measurements.

Remark 9.3: Special case

In the special case of $n_{u_i} = n_{\mathbf{g}}$, $\nabla_{u_i}\mathbf{g}_i\left(u_i, d_i\right)$ is a square matrix.

Coordinator: The coordinator problem is formulated as

$$\min_{\mathbf{t}_0, \ldots, \mathbf{t}_N} \quad \sum_{i=0}^{N} \mathcal{L}_i(u_i, d_i, \mathbf{t}_i, \boldsymbol{\lambda}_i) \tag{9.7a}$$

$$\text{s.t.} \quad \sum_{i=0}^{N} \mathbf{t}_i - \bar{\mathbf{g}} = 0, \tag{9.7b}$$

$$\mathbf{t}_0 \geq 0 \tag{9.7c}$$

where $\boldsymbol{\lambda}_i$ is considered as parameter provided by local Langrange multipliers estimator (9.6). Note that $\mathbf{g}_i$ is equal to $\mathbf{t}_i$ in the steady state (by utilizing local constraint control), and $\mathbf{t}_0$ is non-negative (by incorporating `max` selector in a shown in equation (9.11a).

To solve problem (9.7) using a feedback control loop, one can for reformulate problem (9.7) into the following problem.

$$\min_{\mathbf{t}_0, \ldots, \mathbf{t}_{N-1}} \sum_{i=0}^{N-1} \left[\boldsymbol{\lambda}_i^{\top}\left(\mathbf{g}_i\left(u_i, d_i\right) - \mathbf{t}_i\right) + \lambda_N^{\top}\left(\mathbf{g}_N\left(u_N, d_N\right) - \mathbf{t}_N\right) \right] \tag{9.8}$$

where $\mathbf{t}_N = \bar{\mathbf{g}} - \sum_{i=0}^{N-1} \mathbf{t}_i$ [44].

At stationary condition,

$$\nabla_{\mathbf{t}_i} \left[\sum_{i=0}^{N-1} \left[\boldsymbol{\lambda}_i^{\top}\left(\mathbf{g}_i\left(u_i, d_i\right) - \mathbf{t}_i\right) + \boldsymbol{\lambda}_N^{\top}\left(\mathbf{g}_N\left(u_N, d_N\right) - \mathbf{t}_N\right) \right] \right] = 0 \tag{9.9}$$

Thus,

$$-\boldsymbol{\lambda}_i^{\top} + \boldsymbol{\lambda}_N^{\top} = 0, \qquad i = 0, \ldots, N-1 \tag{9.10}$$

Note that $J_i\left(u_i, d_i\right)$ terms can be omitted, since they are independent of $\mathbf{t}_i$ and are constants.

Problem (9.8) is an unconstrained optimization with N degree of freedom, which can be updated using I-controllers as follows.

$$\mathbf{t}_0^{k+1} = \mathbf{t}_0^k + \max\left(0, K_0\left(-\boldsymbol{\lambda}_0 + \boldsymbol{\lambda}_N\right)\right) \tag{9.11a}$$

$$\mathbf{t}_i^{k+1} = \mathbf{t}_i^k + K_i\left(-\boldsymbol{\lambda}_i + \boldsymbol{\lambda}_N\right),\ i = 1, \ldots, N-1 \tag{9.11b}$$

$$\mathbf{t}_N = \bar{\mathbf{g}} - \sum_{i=0}^{N-1} \mathbf{t}_i \tag{9.11c}$$

where eq. (9.11a) is for virtual subsystem to store unutilized resources, and *Max* Selector is chosen to ensure non-negative storage values. The value of $\boldsymbol{\lambda}_0$ is 0 because the value of J_0 is always constant. Further, eq. (9.11b) is for all $i = 1, \ldots, N-1$, and eq. (9.11c) is for compensator subsystem that guarantees the constraint (primal) feasibility.

9.2.2 Dual-based Distributed Feedback-optimizing System

This structure is based on the method proposed and discussed in [40, 41, 65]. To be able to implement dual-based distributed feedback-optimizing control, the subsystem should satisfy Assumption 9.2.

Introducing the Lagrange function

$$\mathcal{L}\left(\boldsymbol{\lambda}, \mathbf{u}, \mathbf{d}\right) = J\left(\mathbf{u}, \mathbf{d}\right) + \boldsymbol{\lambda}^\top \mathbf{g}\left(\mathbf{u}, \mathbf{d}\right)$$

and defining $\mathbf{g} = \begin{bmatrix} g_1 & \ldots & g_{n_\mathbf{g}} \end{bmatrix}^\top$, and $\boldsymbol{\lambda} = \begin{bmatrix} \lambda_1 & \ldots & \lambda_{n_\mathbf{g}} \end{bmatrix}^\top$, the necessary conditions for optimality (KKT conditions) for the problem (9.1) can be expressed as

$$\nabla_\mathbf{u} \mathcal{L}(\boldsymbol{\lambda}, \mathbf{u}, \mathbf{d}) = \mathbf{0} \tag{9.12a}$$

$$g_i(\mathbf{u}, \mathbf{d}) \leq 0,\ \text{for } i = 1, \ldots, n_\mathbf{g} \tag{9.12b}$$

$$\lambda_i \geq 0,\ \text{for } i = 1, \ldots, n_\mathbf{g} \tag{9.12c}$$

$$\lambda_i g_i(\mathbf{u}, \mathbf{d}) = 0,\ \text{for } i = 1, \ldots, n_\mathbf{g} \tag{9.12d}$$

The unknown variables in equation set (9.12) are $\mathbf{u}$ and $\boldsymbol{\lambda}$. The equation set can be solved using dual ascent [64]. Here we solve (9.12a) with respect to $\mathbf{u}$ with a fixed value of $\boldsymbol{\lambda}$, and then iteratively update $\boldsymbol{\lambda}$ in an outer loop to satisfy the remaining equations, where the most important is to keep $\mathbf{g} = \mathbf{0}$ for the case when the constraints are active, which corresponds to a nonzero $\boldsymbol{\lambda}$.

Subproblems as feedback control problems: Given a fixed value of $\boldsymbol{\lambda}$, the stationary condition (9.12a) is the solution of the following unconstrained problem,

$$q(\boldsymbol{\lambda}) := \min_{\mathbf{u}} \quad \mathcal{L}\left(\boldsymbol{\lambda}, \mathbf{u}, \mathbf{d}\right) \tag{9.13}$$

Assuming additively separable cost and constraint function, the controlled variable

$$\begin{aligned} \mathbf{c}(\boldsymbol{\lambda}) &:= \nabla_\mathbf{u} \mathcal{L}(\boldsymbol{\lambda}, \mathbf{u}, \mathbf{d}) \\ &= \begin{bmatrix} \nabla_{u_1} J_1(\mathbf{u}, \mathbf{d}) \\ \vdots \\ \nabla_{u_N} J_N(\mathbf{u}, \mathbf{d}) \end{bmatrix} + \boldsymbol{\lambda}^\top \begin{bmatrix} \nabla_{u_1} \mathbf{g}(\mathbf{u}, \mathbf{d}) \\ \vdots \\ \nabla_{u_N} \mathbf{g}(\mathbf{u}, \mathbf{d}) \end{bmatrix} \end{aligned} \tag{9.14}$$

can be easily decomposed into each subsystem control as the following

$$\mathbf{c}_i(\boldsymbol{\lambda}) := \nabla_{u_i}\mathcal{L}_i(\boldsymbol{\lambda}, \mathbf{u}, \mathbf{d}) = \nabla_{u_i}J_i(\mathbf{u}, \mathbf{d}) + \boldsymbol{\lambda}^\top \nabla_{u_i}\mathbf{g}(\mathbf{u}, \mathbf{d}) \tag{9.15}$$

Therefore, the only step in the subsystem is controlling $\mathbf{c}_i(\boldsymbol{\lambda})$ to setpoint of $\mathbf{c}_i^{sp} = 0$ by manipulating the primal variable u_i.

Coordinator: The coordinator problem is formulated as

$$\max_{\boldsymbol{\lambda}} \quad q(\boldsymbol{\lambda}) \tag{9.16}$$

Problem (9.16) is an unconstrained optimization with $n_{\mathbf{g}}$ degree of freedom, which can be updated using I-controllers (which is also known as subgradient updates in optimization literatures) as follows.

$$\lambda_i^{k+1} = \texttt{max}\left(0, \lambda_i^k + K_{\lambda_i} g_i(\mathbf{u}, \mathbf{d})\right), \text{ for } i = 1, \ldots, n_{\mathbf{g}} \tag{9.17}$$

`Max` Selector is chosen to ensure non-negative Lagrange multipliers. When the constraint is inactive, `Max` Selector chose zero instead of negative for the associated Lagrange multipliers. This satisfies dual feasibility (9.12c).

9.2.3 Dual-based Distributed Feedback-optimizing System with Override

In the structure above, the constraints are controlled in the outer loop, in the slower timescale. However, in many processes, it is desirable to control the constraints in the faster timescale. To address this, we proposed dual-based approach with override [47]. To be able to implement dual-based distributed feedback-optimizing control with override, the subsystem should satisfy Assumption 9.2.

Subproblems as feedback control problems: Consider subsystem i that is specifically assigned to control constraint g_i in fast time scale, then Subsystem i solves problem (9.13) and control the active constraint using feedback-optimizing control, by executing the following steps.

1. Gradient controller: Control $c_i(\boldsymbol{\lambda})$ to a setpoint of $c_i^{sp} = 0$ by manipulating *calculated* primal variable $\tilde{u}_i$.
2. Override constraint controller: Control coupling constraint $g_i(\mathbf{u}, \mathbf{d})$ to a setpoint of 0 by manipulating *calculated* primal variable u_i^g.
3. Select implemented primal variable u_i: If small value of primal variable is good for constraint satisfaction, then $u_i = \texttt{min}\left(\tilde{u}_i, u_i^g\right)$. Otherwise, $u_i = \texttt{max}\left(\tilde{u}_i, u_i^g\right)$.

Coordinator: For the constraint that is overridden (when it's active) in the subsystem, an auxiliarry constraint is introduced as follows,

$$\tilde{g}_i = \tilde{u}_i - u_i^g \tag{9.18}$$

which is controlled by manipulating the corresponding Lagrange multipliers. This can be done using I-controllers as follows.

$$\lambda_i^{k+1} = \texttt{max}\left(0, \lambda_i^k + K_{\lambda_i}\tilde{g}_i\right) \tag{9.19}$$

Similarly, `Max` Selector is chosen to ensure non-negative Lagrange multipliers. This satisfies dual feasibility (9.12c).

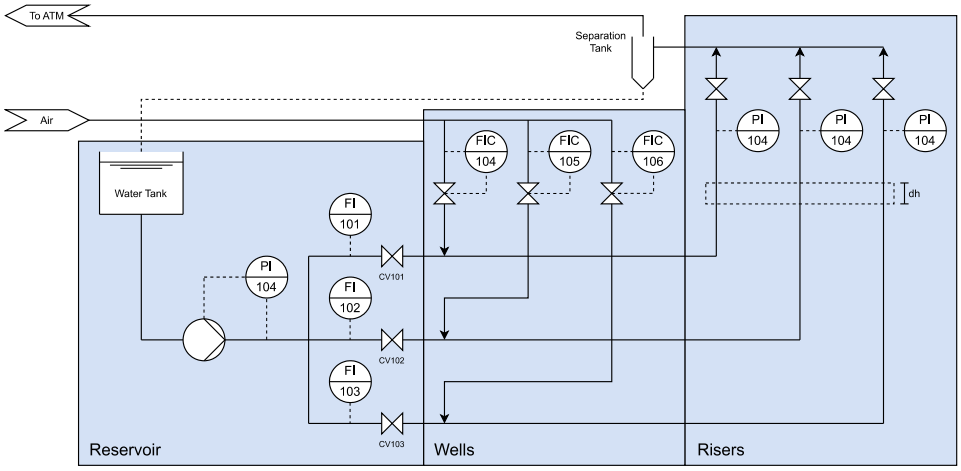


Figure 9.1: This experiment's schematic is based on [83].

9.3 Experimental Setup

Subsea production systems involve drilling wells on the seabed to extract hydrocarbons from underground reservoirs. These hydrocarbons are transported in pipelines along the seabed and lifted to the surface via a riser pipeline. However, if the reservoir pressure is low, artificial lift methods, such as gas-lift, may be necessary to overcome pressure losses and bring the hydrocarbons to the surface. In gas-lift, compressed gases are injected via annulus into the well tubing to reduce the fluid mixture density and minimize hydrostatic pressure losses. It's important to note that excessive gas injection can increase frictional pressure drop in the well tubing, counteracting the desired effect [60]. As such, each well has a local optimum gas injection rate. Since the total lift gas available is often limited, it must be optimally allocated among wells to maximize production across the network. For further information on production optimization in the oil and gas industry, please refer to [14, 84], and related literature.

9.3.1 Experimental rig as a subsea production system

In order to emulate a subsea gas-lifted oil production system, a laboratory-scale experimental rig is utilized. The rig operates using water and air as working fluids instead of oil and gas for simplicity. The choice of working fluids does not affect the gas lift phenomenon, which can still be observed in the lab rig. Therefore, the rig is suitable for studying production optimization techniques, where the gas lift effect is the phenomenon of interest. The schematic of the system, which includes a reservoir, well, and riser section, is shown in Figure 9.1. The system measurements, denoted as $\mathbf{y_p}$, include the well top pressures (PI101, PI102, and PI103), the pump outlet pressure (PI104), the liquid flowrates (FI101, FI102, and FI103), and the gas flowrates (FI104, FI105, and FI106). To control

the gas flowrates, denoted as $\mathbf{u} = \begin{bmatrix} Q_{gl,1} & Q_{gl,2} & Q_{gl,3} \end{bmatrix}^\top$ to the calculated setpoints denoted as $\mathbf{u}^{sp} = \begin{bmatrix} Q^{sp}_{gl,1} & Q^{sp}_{gl,2} & Q^{sp}_{gl,3} \end{bmatrix}^\top$, three PI controllers are used. The reservoir valve openings (CV101, CV102, and CV103) represent system disturbances and vary during the experiments to simulate different reservoir behaviors. A PI controller maintains a constant pump outlet pressure throughout the experiment.

The reservoir section is comprised of a stainless steel tank, a centrifugal pump, and three control valves (CV101, CV102, and CV103). These valves are utilized to mimic disturbances from the reservoir, such as pressure oscillations or reservoir depletion. With this setup, the reservoir produces only liquid with outflow rates ranging from 2 L/min to 15 L/min. Flow meters (FI101, FI102, and FI103) are installed before the reservoir valves to measure the outflow rates. A PI controller is used to adjust the pump rotation and maintain the pump's outlet pressure (PI104) at a constant 0.3 barg in this experiment.

The well section consists of three parallel flexible hoses with 2 cm inner diameters and a length of 1.5 m. Pressurized air at approximately 0.5 barg is injected by three air flow controllers (FIC104, FIC105, and FIC106) approximately 10 cm after the reservoir valves, within the range of 1 sL/min to 5 sL/min.

The riser section consists of three vertical pipelines, orthogonal to the well section, with 2 cm inner diameters and a height of 2.2 m. The pressures on top of the risers (PI101, PI102, and PI103) are measured. Three manual valves are kept open during the experiments after the sensors. The air is vented out to the atmosphere, and the liquid is recirculated to the reservoir water tank. Further details on the experimental setup can be found in [83].

9.3.2 Optimization problem setup

In this experimental setup, the optimization problem aims to maximize the network liquid flow rate, which is the combined production of liquid from three wells, while taking into account a limited amount of gas-lift injection. To express the economic objectives in line with problem (9.1), we can state the following:

$$\begin{aligned} J(\mathbf{u}, \mathbf{p}) &:= \sum_{i=1}^{3} f_i(u_i, p_i) = \\ &- 20Q_{l,1}(u_1, p_1) - 25Q_{l,2}(u_2, p_2) - 30Q_{l,3}(u_3, p_3) \end{aligned} \tag{9.20}$$

where $Q_{l,1}$, $Q_{l,2}$, and $Q_{l,3}$ are the produced liquid flowrates of wells 1, 2, and 3, respectively. For illustration, we assume that the wells have different hydrocarbon prices as shown above. The input vector is given by

$$\mathbf{u} = \begin{bmatrix} Q_{gl,1} & Q_{gl,2} & Q_{gl,3} \end{bmatrix}^\top$$

where $Q_{gl,1}$, $Q_{gl,2}$, and $Q_{gl,3}$ are the injected gas flowrates of wells 1, 2, and 3, respectively.

Regarding optimization, the injected gas flowrates serve as the decision variables. For the plant, however, they represent setpoints to be tracked. The experimental lab rig depicted in Fig. 9.1 employs flow indicator and controllers (FICs) 104, 105, and 106 to maintain the air injection flowrates at their respective setpoints. As a result, we denote these decision variables as $\mathbf{u}^{sp} = \begin{bmatrix} Q^{sp}_{gl,1} & Q^{sp}_{gl,2} & Q^{sp}_{gl,3} \end{bmatrix}^\top$. Some may suggest using valve opening of

the FICs as the decision variables. However, this alternative may encounter practical issues due to valve non-linearity, and hysteresis behavior. One crucial point to note is that this alternative formulation renders immediate decomposition, as illustrated in equation (9.14), impossible. Additionally, three components of $\mathbf{p}$, which correspond to reservoir valve openings CV101, CV102, and CV103, fluctuate over time. This implies that the cost is also a function of $\mathbf{p}$.

The total gas availability, which is a shared (input) constraint, can also be expressed as follows:

$$\begin{aligned} g(\mathbf{u}, \mathbf{p}) &:= \sum_{i=1}^{3} g_i(u_i, p_i) - g^{max} \\ &= Q_{gl,1} + Q_{gl,2} + Q_{gl,3} - Q_{gl}^{max} \end{aligned} \tag{9.21}$$

where we directly measure the constraint, and we use FICs to drive $Q_{gl,i}$ to $Q_{gl,i}^{sp}$.

Remark 9.4: Type of constraint

Note that the total gas availability in general is an inequality constraint. However, in this experimental setup, we found that the gas lift constraint is always active, (which is also common in many gas-lifted oil fields). Hence we can equivalently consider the coupling constraint to be an equality constraint, and then we do not need the virtual subsystem (9.11a) and the max selector (please refer to eq. (9.17) and (9.19)) for $\boldsymbol{\lambda}$ update.

9.4 Control Setup of Distributed Feedback-optimizing System

Here, we establish a distributed feedback-optimizing control structure for our experimental setup, with each well representing a subsystem. These subsystems employ local gradient estimators based on forward sensitivity analysis to evaluate the local cost gradient and constraint gradient, supported by a reliable dynamic model. Estimating the system's current states, both differential and algebraic, is achieved via an extended Kalman filter (EKF) in each subsystem, given accurate state estimation and proper noise filtering.

Control Tuning Parameter Selection: Recommended Practice The coordinator controller operates on a slow timescale, while the local controller on a fast timescale, typically chosen with a ratio between 5 to 10 [57]. We use PI (Proportional Integral) and I (Integral) controllers tuned using SIMC-rules [52].

Controller tuning validation Prior to the implementation in the rig, we validated the controller tunings using a high-fidelity dynamic MATLAB model that includes lower-layer controller dynamics, i.e. FICs, and tuned noise parameters based on rig data. Detailed parameters are available on our Github page[1], and a comprehensive model description can be found in [83].

[1] https://github.com/Process-Optimization-and-Control/ProductionOptRig

Remark 9.5: Tuning Validation

The simulator model is solely utilized for controller tuning parameter determination, preceding its implementation on the physical rig. All results in Section 5 originate from experiments conducted on the actual rig, not the simulator.

Steady-state gradient estimation This work uses forward sensitivity analysis to estimate the steady-state gradient which also has been described in 5.2.3.

9.4.1 Primal-based Distributed Feedback-optimizing Control Setup

In this experiment, individual subsystems do not require local PI controllers as we consider shared input constraints. Coordinator/central controllers, responsible for equalizing all Lagrange multipliers, generate setpoints, denoted as $Q_{gl,i}^{sp}$, which are used by the flow controller FIC$_i$ to regulate air injection valve openings. This eliminates the need for additional layers beyond the local setpoint controller.

For the coordinator controllers, we use integral controllers, with integral gains denoted as $K_{I,i} = \frac{1}{K_{\lambda,i}(\tau_{c,i}+\theta_i)}$, where i represents the well index. Here, $K_{\lambda,i}$ and θ_i denotes the step response and time delay of the constraints determined by local Lagrange multipliers, and $\tau_{c,i}$ is the closed-loop time constants.

The parameters $K_{\lambda,i}$ and θ_i are determined by analyzing the step response. Initially, setting $\tau_{c,i} = 1$ is ideal for achieving fast steady-state, but considering the coordinator controller's timescale should be slower than the plant, adjusted parameters are chosen based on observation and practical considerations.

Given the equality constraint, there's no requirement for a virtual subsystem (see Remark 9.3). Nevertheless, selecting a compensator subsystem remains necessary.

Compensator Subsystem Selection The Compensator subsystem is selected if its accummulated profit difference to the other subsystem at the steady-state is positive. The difference, denoted as $P_c(t)$, is calculated using the following formula,

$$P_{d,c}(t) = \frac{P_c(t) - P_i(t)}{P_i(t)}.100 \tag{9.22}$$

where $P_c(t)$ represents the accumulated profit with the candidate subsystem as the compensator, and $P_i(t)$ represents the accumulated profit with subsystem i as the compensator. The accumulated profit values are determined using a MATLAB-developed lab rig model.

According to this metric, subsystem three (3) emerges as the optimal compensator candidate. This result is expected due to well 3 having the most expensive oil, see eq. (5.1). The implemented structure is depicted in Figure 9.2, and the controller and tuning parameters for the primal-based scheme can be found in Table 9.1.

9.4.2 Dual-based Distributed feedback-optimizing Control Setup

The local gradient controller generates the setpoint for the PI flow controller FIC$_i$, regulating the air injection valve opening. For the central constraint controller, the constraint **g** is directly measured to update to the Lagrange multiplier, essential for calculating controlled variables in the local gradient controller. This necessitates the addition of *two more layers* above the local setpoint controller.

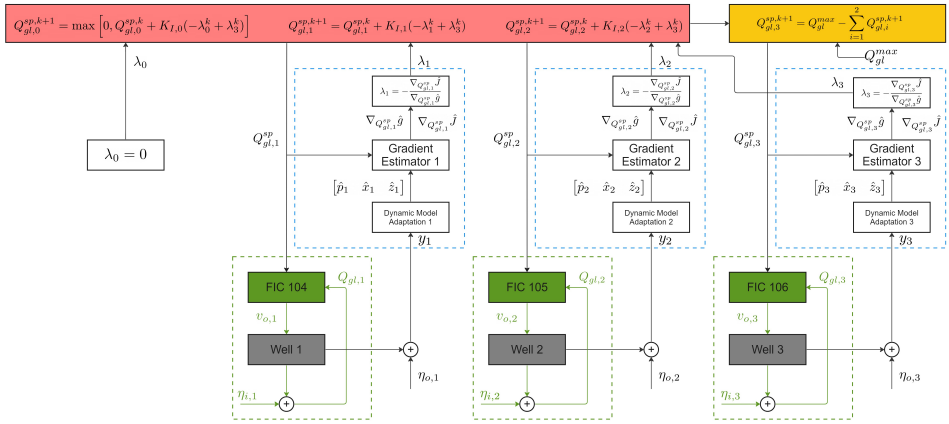


Figure 9.2: Primal-based distributed feedback-optimizing control implemented in the experimental lab rig [91].

For the coordinator/central constraint controller, the integral gain ($\boldsymbol{\alpha}$) is determined as $\frac{1}{K_\lambda(\tau_{\lambda,c}+\theta_\lambda)}$, where K_λ and θ_λ pertain to the constraint's step response and time delay, while $\tau_{\lambda,c}$ governs $\mathbf{g}$'s evolution.

For the three local gradient controllers (indexed by $i = 1, 2, 3$), the integral gain ($K_{I,i}$) is calculated as $\frac{1}{K_{\mathbf{u}_i}(\tau_{\mathbf{u}_i,c}+\theta_{\mathbf{u}_i})}$, where $K_{\mathbf{u}_i}$ and $\theta_{\mathbf{u}_i}$ correspond to the gradient's step response and time delay with respect to the primal variables, and $\tau_{\mathbf{u}_i,c}$ governs the evolution of $c_i(\boldsymbol{\lambda})$.

We determine K_λ, θ_λ, $K_{\mathbf{u}_i}$, and $\theta_{\mathbf{u}_i}$ by analyzing step responses, while considering the concept of time scale separation for $\tau_{\mathbf{u}_i,c}$ and $\tau_{\lambda,c}$. Similarly, we adjust $\tau_{u_i,c}$ based on observed results and practical justification.

We avoid overly aggressive control settings. Additionally, we acknowledge that local gradient controllers are not at the lowest hierarchy level (Fig. 9.3), necessitating a slower timescale than the plant. Refer to Fig. 9.3 for the lab-rig's control structure, and find the controller and tuning parameters for the dual-based scheme in Table 9.1.

9.4.3 Dual-based Distributed feedback-optimizing Control Setup with Override

In this structure, the local gradient controller also generates the setpoint for the PI flow controller FIC$_i$, regulating the air injection valve opening. Additionally, the central constraint controller controls the *auxiliary* constraint $\tilde{\mathbf{g}}$ to update the Lagrange multiplier. A selected subsystem employs an override scheme, wherein the fast control of the actual constraint $\mathbf{g}$ (typically faster than the local gradient controller), leading to the override of the output ($Q^{sp}_{gl,i}$) by the override constraint controller's result.

In addition to the coordinator and local gradient controller, the dual-based with override scheme requires an override constraint controller. The integral gain, denoted as $K_{I,g}$, is calculated as $K_{I,g} = \frac{1}{K_{g,i}(\tau_{g,i}+\theta_{g,i})}$, where $K_{g,i}$ and $\theta_{g,i}$ relate to the step response and time delay of the constraint with respect to the chosen primal variable (u_i), and $\tau_{g,i}$ gov-

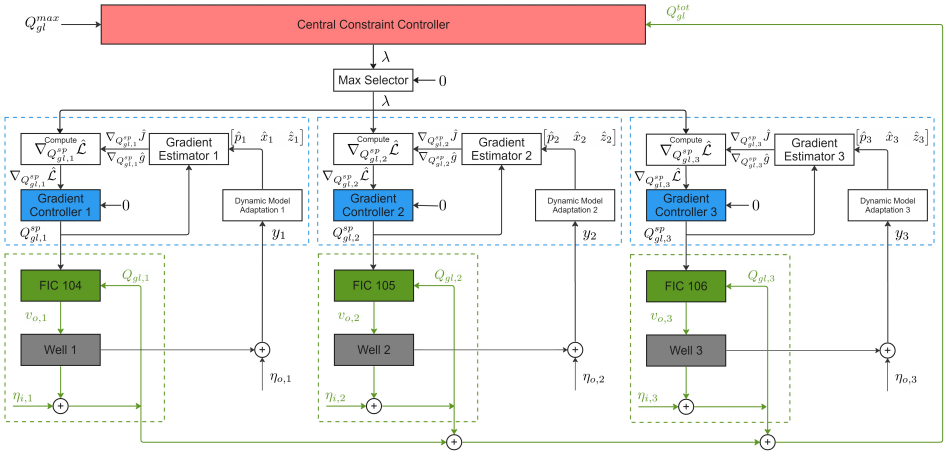


Figure 9.3: Dual-based distributed feedback-optimizing control implemented in the experimental lab rig [65].

erns the evolution of g. The concept of determining and tuning the controller parameter aligns with the prior controller explanation.

Anti-windup To choose between the control actions generated by the gradient controller and the override constraint controller, a selector is employed, allowing only one controller's output to influence the plant at any given moment. This selective action can lead to integral term accumulation, known as windup, in the controllers not currently in use. To mitigate this issue, this paper adopts a back-calculation scheme proposed by [51] and suggests an appropriate anti-windup gain value of $K_{aw} = \frac{1}{\tau_{1,g}}$, with $\tau_{1,g}$ representing the time constant for the override constraint controller.

Gain Scheduling To mitigate the deceleration of the dual variable trajectory due to active constraint satisfaction through override control, it is essential to incorporate a proportional gain $(K_{P,\alpha})$ into the central constraint controller. This gain is computed as follows:

$$K_{P,\alpha} = \begin{cases} \frac{\tau_{1,\alpha}}{K_\lambda(\tau_{\lambda,c}+\theta_\lambda)}, & \text{if } l = 1 \\ 0, & \text{otherwise} \end{cases} \tag{9.23}$$

Here, $\tau_{1,\alpha}$ denotes the time constant, and l denotes the status of the override constraint controller, with $l = 1$ denoting selection of the control action of the override constraint controller, and $l = 0$ otherwise.

Override Subsystem Selection There are three potential configurations for a subsystem to involves an override scheme due to one constraint and three subsystems. The selection process involves ensuring that the chosen configuration avoids local input constraints (see Section 9.3.1) to maintain simplicity in the problem formulation. Additionally, the configuration with the best constraint satisfaction, often assessed using the maximum constraint violation magnitude, is preferred, with the selection process facilitated using a MATLAB-based lab rig model. Based on this selection strategy, we found that subsystem three (3) is the best candidate to contribute in override scheme.

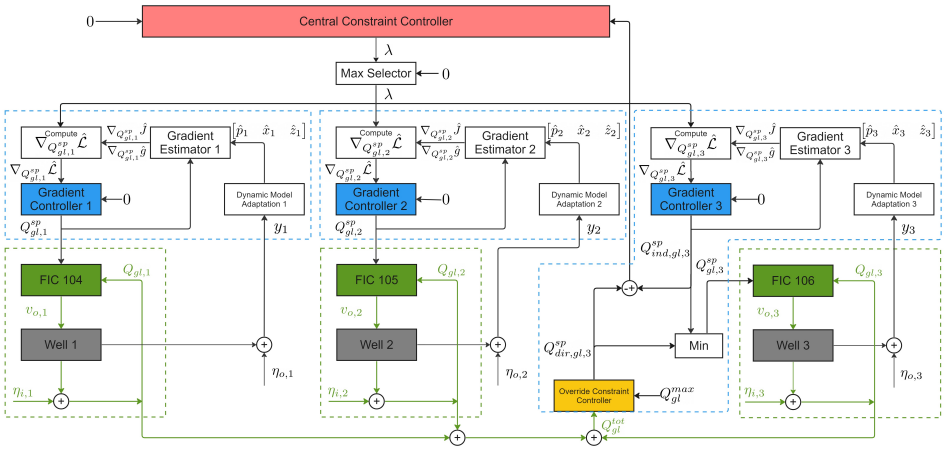


Figure 9.4: Dual-based distributed feedback-optimizing control with override implemented in the experimental lab rig [47].

Given the selected override configuration, and based on the description in Section 2.3, Figure 9.4 illustrates the implemented structure, and Table 9.1 contains the obtained controller and tuning parameters for Dual-based with override.

9.5 Experimental Results

Utilizing the control and tuning parameters shown in Table 9.1, we implemented primal-based, dual-based, and dual-based with override scheme. Figure 9.5 shows the reservoir valve openings ($CV101$, $CV102$, $CV103$) that we consider as the disturbance in this experiment. The first disturbance occurs when the opening of $CV101$ gradually decreases from $t = 6.5$ to $t = 14$ minutes. We expect a decrease in the gas-lift injection in well 1, and a redirection of the gas supply to the other wells. The second disturbance occurs when the opening of $CV103$ also gradually decreases from $t = 15.5$ to $t = 21$ minutes. We expect that the gas supply to well 3 reduces with larger rate since the "hydrocarbon price" of this well is higher. Meanwhile the other wells will obtain more gas supply with larger rate as well. The third disturbance occurs when the opening of $CV103$ gradually increases from $t = 24$ to $t = 29.5$ minutes. We expect a reverse reaction to the reaction of the second disturbance. Finally, the fourth disturbance occurs when the opening of $CV101$ also gradually increases from $t = 33$ to $t = 42$ minutes. Similarly, we expect a reverse reaction to the reaction of the first disturbance. We try to avoid sudden disturbance to ensure that the controller can adjust the plant smoothly.

In the rig, we used a programming environment (LABVIEW [87]) to automate the implementation of these disturbance. Therefore, it is possible to repeat the independent experiments with the same disturbance profile. The following results are the average of three independent experiments.

Comparison of the optimal setpoints We run the experimental results comparing primal-based, dual-based, dual-based with back-off and dual-based with override scheme, which are presented in Figure 9.6-9.8. Figure 9.6 depicts the actual gas-lift flow rate. The meas-

Table 9.1: Controller and Tuning parameters

Description	Variable	Value
Exp. rig sensors sampling time	T_s	1 second
Primal-based		
Coordinator 1 I-Gain	$K_{I,1}$	-0.016
Coordinator 2 I-Gain	$K_{I,2}$	-0.014
Coordinator 3 I-Gain	$K_{I,3}$	-0.011
Dual-based		
Coordinator I-Gain	α	0.0088
Gradient Controller Input 1 I-Gain	$K_{I,1}$	0.016
Gradient Controller Input 2 I-Gain	$K_{I,2}$	0.014
Gradient Controller Input 3 I-Gain	$K_{I,3}$	0.011
Dual-based with Override		
Coordinator P-Gain	$K_{P,\alpha}$	1.7686
Coordinator I-gain	α	0.00368
Gradient Controller Input 1 I-Gain	$K_{I,1}$	0.016
Gradient Controller Input 2 I-Gain	$K_{I,2}$	0.014
Gradient Controller Input 3 I-Gain	$K_{I,3}$	0.011
Override Controller I-Gain	$K_{I,3}$	0.1
Override Controller AntiWindup-Gain	K_{aw}	0.4
Local set point controllers		
FIC 104 P-Gain	$K_{P,FIC,1}$	8560
FIC 105 P-Gain	$K_{P,FIC,2}$	8560
FIC 106 P-Gain	$K_{P,FIC,3}$	8560
FIC 104 I-Gain	$K_{I,FIC,1}$	100
FIC 105 I-Gain	$K_{I,FIC,2}$	100
FIC 106 I-Gain	$K_{I,FIC,3}$	50

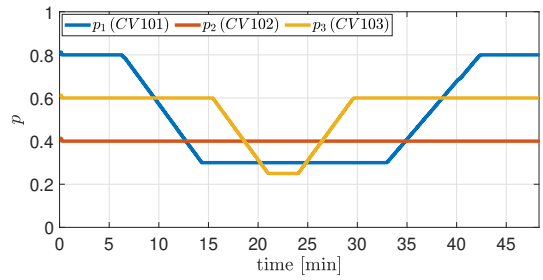


Figure 9.5: The change of reservoir valve openings (CV101, CV102, and CV103) during the experiments for representing different reservoir behaviors. These reservoir valve openings are system disturbances.

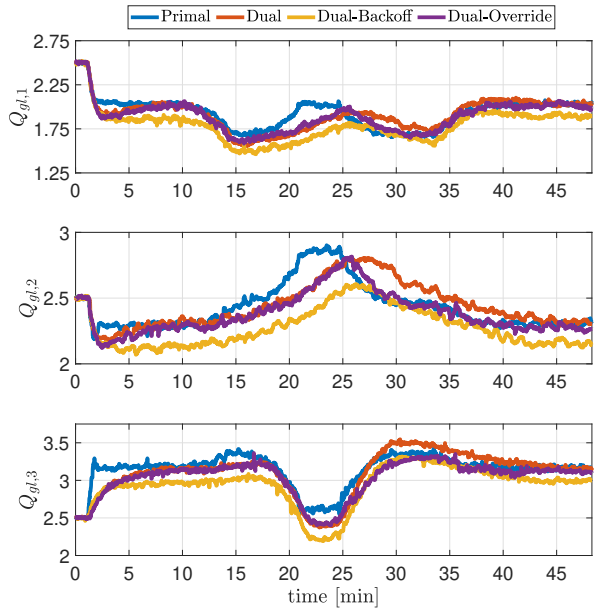


Figure 9.6: The measured gas-lift flow rate ($\mathbf{u}$) of every wells due to reservoir parameter changing (disturbance) from the Experimental Lab Rig.

ured trajectories are slightly different from the calculated input setpoint shown in Figure 9.7. This difference occurs due to input measurement noise and the fact that the gas flowrates controllers (FIC 104, FIC 105, FIC 106, see Figure 9.1) need time to settle the actual gas-lift flow rate $\mathbf{u}$ to the setpoint of gas-lift flow rate $\mathbf{u}^{sp}$.

Figure 9.6-9.7 show that the dual-based are relatively slower in responding to the second disturbance (see around $t = 25$ min). This slow response is the consequence of considering time-scale separation concept.

Comparison of constraint satisfaction As can be seen in the top plot of Figure 9.8, due to time-scale separation, the dual-based scheme significantly violates the constraint. One may think that incorporating Proportional-gain can be viable solution. Unfortunately, that idea leads to even worse constraint violation because the central constraint controller becomes more aggressive. For this specific case, primal-based scheme appears to be the best in satisfying the constraint compared to all dual-based schemes. This is the result of the presence of the compensator subsystem. Meanwhile, dual-based scheme seems requiring back-off to satisfy the constraint better, and relatively slow to recover. Dual-based with override scheme seems performing pretty well in satisfying the constraint compared to dual-based scheme. It may still relatively slow to reach the active constraint, but the presence of override control and gain scheduling have shown significant result in satisfying the constraint (see after $t = 25$ minutes). Thus, back-off parameter may be insignificant or even unnecessary.

The bottom plot of Figure 9.8 shows the associated Lagrange multiplier for the dif-

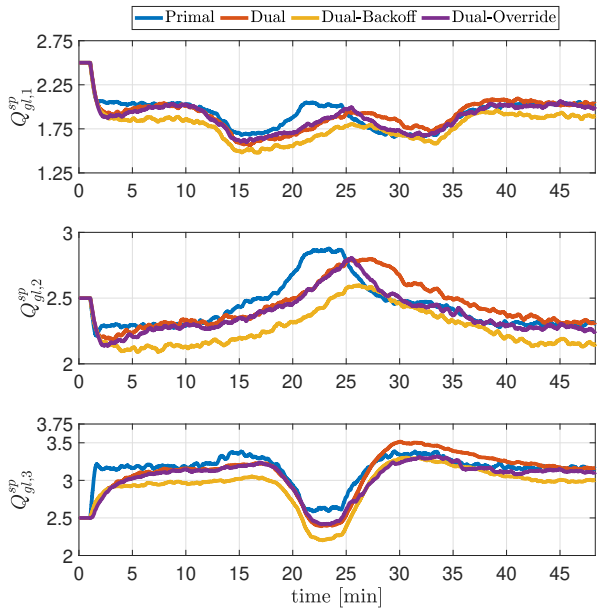


Figure 9.7: The gas-lift flow rate setpoint $\left(\mathbf{u}^{sp} = \mathbf{Q}^{sp}_{gl}\right)$ of every wells due to reservoir parameter changing (disturbance) from the Experimental Lab Rig.

ferent schemes. It is important to notice that all the three local Lagrange multipliers of the primal based converge to the same trajectory, and in general, they are faster than all dual-based schemes. Meanwhile, each dual-based scheme has one Lagrange multiplier, and they are slower as they are controlled in the slow time scales. It is interesting to notice that before $t = 25$ minute, dual-based with override has similar performance (trajectory) to dual-based scheme. However, after $t = 25$ minute, dual-based with override has quite similar performance (trajectory) to primal-based scheme. Specifically at the period between $t = 25$ to $t = 35$ minute, we can notice that dual-based with override is "enforcing" the trajectory (both the constraint and the associated dual variable) in order to minimize the constraint violation (in terms of magnitude and/or duration).

Comparison of the optimal cost To analyze the optimization performance of primal-based, dual-based with back-off and dual-based with override, we compare the profit (from both individual experiments) obtained by the three schemes with the naive approach, where we consider fixed inputs, i.e.,

$$\mathbf{u} = \begin{bmatrix} Q^{sp}_{gl,1} & Q^{sp}_{gl,2} & Q^{sp}_{gl,3} \end{bmatrix}^\top = \begin{bmatrix} \frac{Q^{max}_{gl}}{3} & \frac{Q^{max}_{gl}}{3} & \frac{Q^{max}_{gl}}{3} \end{bmatrix}^\top$$

The naive approach illustrates the case in which no information about the system is available. Hence, the best alternative is to divide the available gas equally among the wells. This latter approach is another benchmark to show that the optimization schemes, will give more profit compared to not doing any optimization at all.

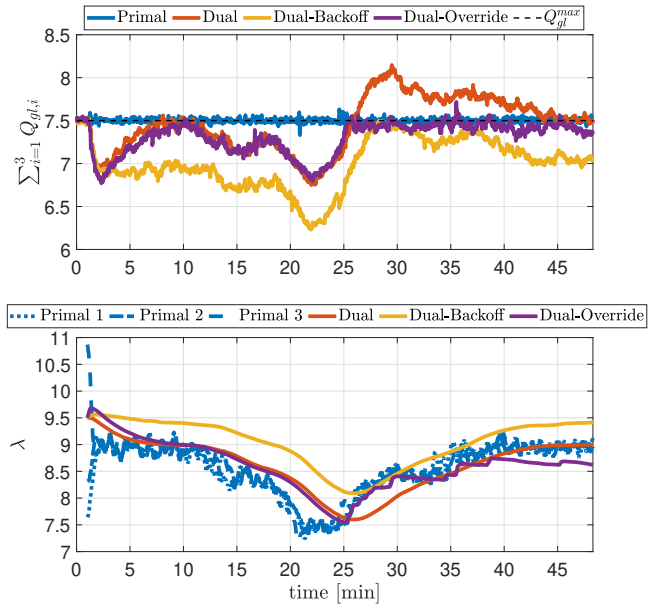


Figure 9.8: Constraint satisfaction and Lagrange multiplier evolution due to reservoir parameter changing (disturbance) from the Experimental Lab Rig.

Note that we do not include dual-based in comparison as it significantly violates the constraint, which implies that it has significant trajectory in infeasible region. In other words, dual-based requires significant extra resource starting from $t = 25$ min.

To measure the performance, we plot the difference, in percentage, between the instantaneous profit of the approach of interest and the naive approach. The difference is calculated as

$$P_{diff} = \frac{P - P_{naive}}{P_{naive}}.100 \tag{9.24}$$

where P is the profit of the approach of interest, and P_{naive} is the profit of the naive approach.

Figure 9.9 depicts the cumulative profit/loss difference $(\sum P_{diff})$ of all the three schemes of interest (i.e., primal-based, dual-based with back-off, and dual-based with override) to the naive strategy. Figure9.9 shows that the primal-based scheme has the highest cumulative profit difference, and followed by dual-based with override scheme. Interestingly, due to incorporating back-off, dual-based scheme has less cumulative profit compared to the naive approach.

Performance consistency To ensure validity of the results, we then re-run the experiments three times with the same set of disturbance profile for the different approaches. This confirms that the observation discussed earlier is valid.

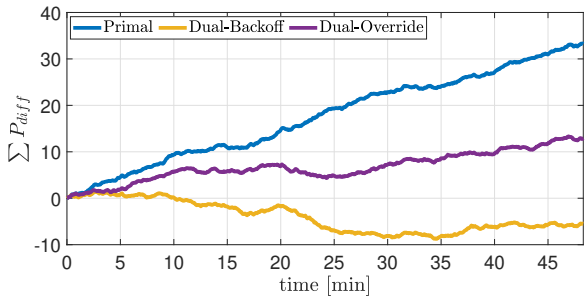


Figure 9.9: The cumulative profit difference from the Experimental Lab Rig.

9.6 Discussion

The presence of local constraint: The primal-based approach entails a relatively strong assumption, as indicated in Assumption 9.1. Specifically, this scheme is not designed to handle saturated inputs, such as when a valve reaches its maximum or minimum opening. In practice, if saturated inputs or local constraints are present, additional strategies must be employed to effectively implement the primal-based scheme. In contrast, any form of dual-based scheme, whether with or without an override, does not suffer from the limitations imposed by local constraints. These constraints can be treated as coupled constraints, but the associated coordinator is only distributed to the affected subsystem. If the constraint solely affects the subsystem in which it is located, it is considered a local constraint. This characteristic distinguishes the dual-based approach, which proves to be more versatile and adaptable in the presence of various types of constraints within a distributed feedback-optimizing control system.

The generation of Lagrange multipliers: The primal-based approach estimates local Lagrange multipliers by evaluating a matrix calculation as depicted in equation (9.6). However, this method relies on the strong assumption that the local Lagrange multipliers are both unique and exist. In contrast, the dual-based approach, whether with or without an override, estimates the Lagrange multipliers through feedback mechanisms, thereby circumventing the need for extensive matrix calculations. As a result, the dual-based method exhibits a reduced likelihood of encountering numerical issues, offering a more robust alternative in more general practical applications.

The number of coupling constraint: According to Assumption 9.1, the primal-based approach is capable of handling a maximum number of coupling constraints equal to or less than the minimum number of local manipulated variables ($n_g \leq n_{u_i}$). In contrast, the dual-based approach, with or without an override, can handle a maximum number of coupling constraints equal to or less than the total number of available manipulated variables ($n_g \leq n_u$). This fundamental distinction implies that the dual-based scheme has the advantage of being able to accommodate a larger number of active constraints compared to its primal-based counterpart.

Additional benefit of the proposed feedback-optimizing schemes In the introduction, we presented 5 benefits of the proposed schemes. When writing this paper, we find that an

additional benefit is that the the dual-based schemes can easily handle linear cost and/or constraints functions. To illustrate this, consider a cost function: $J(u) = 2u$ along with a linear constraint. Consequently, the gradient of the Lagrange function can be expressed as: $\nabla_u \mathcal{L}(u) = 2 + c$. In this case, attempting to solve the equation $\nabla_u \mathcal{L}(u) = 0$ (in order to fulfill the stationary condition) becomes unfeasible, as the decision variable u does not exist after the first differentiation. In contrast, by using feedback, an initial guess of the decision variables is provided and updated throughout the closed-loop system [97].

9.7 Chapter Summary

In this work, we have done experiments to validate all the three distributed feedback-optimizing control schemes. Based on the experiments we can conclude that

- All the three schemes are able to optimally allocate the shared resource. The transient behaviour is slightly different which is affected by the control structure and the choice of the controller tuning parameters.
- In the dual-based schemes, it is necessary to consider the timescale separation between the gradient and constraint controllers. If the central constraint controller is tuned to be in the same timescale as the gradient controllers, then it can lead to instability or oscillatory behavior. However, if the central constraint controller is tuned to be too slow, then the convergence to the optimal steady-state can be too slow. As a consequence, constraint violation is inevitable. Therefore, incorporating back-off is necessary to avoid or at least to minimize constraint violation, leading to steady-state losses.
- Alternatively, this can be addressed by incorporating override constraint control combined with gain scheduling, which improved the constraint handling.
- In this specific case, primal-based scheme is the best among the three schemes. However, primal-based scheme is limited by Assumption 9.1. Considering a more 'general' case, dual-based scheme with override would be the recommended approach.

Part III: Addressing Practical Issues

Chapter 10

Graph-based Primal-based DFoS Framework: Eliminating Coordinator

Both primal-and dual-based distributed optimization system require a master coordinator. However, the use of this coordinator may introduce additional practical challenges, such as impartiality issues, or additional operating costs. This chapter discusses graph-based primal-based dFoS *framework as an alternative solution. This chapter is based on the work in [84].*

10.1 Introduction

Operating a subsea oil production network entails significant challenges and risks, necessitating cooperation among multiple organizations or operator companies. From exploration to exploitation, remote reservoir location discovery is common, often making local processing platform construction economically unviable. Subsea tie-ins, with shared processing facilities, offer a potential solution, utilizing multiple subsea clusters operated by diverse entities with varied local objectives. However, the dispersed locations of these clusters may render centralized coordination economically impractical.

An alternative approach to eliminate the necessity of a master coordinator involves direct coordination among subsystems via a fixed communication network, further reducing information exchange by limiting data communication to a few subsystems rather than broadcasting to a master coordinator [98]. A similar cooperative game approach was employed in interconnected Model Predictive Controller (MPC) in [99], where each MPC collaborates with others through a communication channel to determine optimal input set-points for reference tracking. Cooperative game models have also been proposed to ascertain optimal set-points or trajectories of shared resources, gradually achieving system-wide optimal set-points via neighborhood interactions [100].

Expanding on the work in chapter 8, this chapter explores a graph-based (or consensus-based) decentralized real-time optimization for oil and gas production networks. Here,

subsystems collaborate by negotiating coupling constraints via a fixed communication channel. Information exchange is restricted to directly connected agents, with the negotiated constraints utilized in solving local subproblems, thus achieving system-wide optimization. These local problems may be solved numerically or through feedback mechanisms as detailed in Chapter 8.

10.2 Problem Formulation

Consider a generic optimal resource sharing problem in N different subsystems.

$$\min_{x_1,\dots,x_N} \quad \sum_{i=1}^{N} f_i(x_i) \tag{10.1a}$$

$$\text{s.t.} \quad \sum_{i=1}^{N} A_i x_i - \bar{\mathbf{x}} \leq 0 \tag{10.1b}$$

where $x_i \in \mathbb{R}^{n_{x_i}}$ denotes the decision variables for subsystem i and n_{x_i} is the number of decision variables in subsystem i, $A_i \in \mathbb{R}^{m_x \times n_{x_i}}$ is a matrix that couples different subsystems, $\bar{\mathbf{x}} \in \mathbb{R}^{m_x}$ is the shared resource constraints, m_x is the number of shared resource constraints, and $f_i : \mathbb{R}^{n_{x_i}} \to \mathbb{R}$ is a function that denotes the local objective of subsystem i. Without loss of generality $x_i > 0$ implies that the shared resources is consumed by subsystem i.

Remark 10.1: Local constraints

Each subsystem i may also have local constraints that are assumed to be locally managed by each organization, and are not explicitly shown in the problem formulation (10.1).

Defining $\mathbf{x} = \{x_1, \dots, x_N\}$, the Lagrangian of problem (1) is as follows.

$$\mathcal{L}(\mathbf{x}, \boldsymbol{\lambda}) = \sum_{i=1}^{N} f_i(x_i) + \boldsymbol{\lambda}^\top \left(\sum_{i=1}^{N} A_i x_i - \bar{\mathbf{x}} \right) \tag{10.2}$$

where $\boldsymbol{\lambda} \in \mathbb{R}^{m_x}$ is the Lagrange multiplier of the shared resource constraints.

The objective of the problem (10.1) is to determine optimal shared resource allocation in order to achieve system-wide steady-state optimal operation in a decentralized fashion with limited information sharing over a fixed communication network.

This can be described by an undirected graph $\mathcal{G} = (\mathcal{S}, \mathcal{E}, \mathcal{V})$ with the set of subsystems $\mathcal{S} = \{1, 2, \dots, N\}$, set of communication channel, $\mathcal{E} \subseteq \mathcal{S} \times \mathcal{S}$ and, an adjacency matrix $\mathcal{V} = [v_{i,j}]$. The communication channel of $\mathcal{G}$ is denoted by $e_{i,j} = \{i, j\}$, meaning that there exists a data transmission between subsystems i and j. Since we consider an undirected graph, $e_{i,j} = e_{j,i}$. The adjacency elements associated with the communication line of the graph are positive, i.e. $e_{i,j} \in \mathcal{E} \Leftrightarrow v_{i,j} = 1$. Consequently, for $e_{i,j} \notin \mathcal{E} \Leftrightarrow v_{i,j} = 0$, and we consider $v_{i,i} = 0$ for all i.

The set of neighboring connected plant of subsystem i is defined as follows.

$$\mathcal{N}_i = \{j \in \mathcal{S} : e_{i,j} = (i, j) \in \mathcal{E}\} \tag{10.3}$$

In other words, the set $\mathcal{N}_i$ represents the set of subsystems that directly communicates with subsystem i. Figure 10.1(b) illustrates the definition of neighboring subsystems.

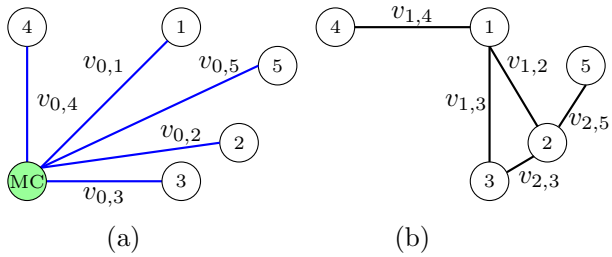


Figure 10.1: (a) Centralized distributed RTO with central coordinator (MC), (b) Decentralized RTO, where $\mathcal{N}_1 = \{2, 3, 4\}, \mathcal{N}_2 = \{1, 3, 5\}, \mathcal{N}_3 = \{1, 2\}, \mathcal{N}_4 = \{1\}$, and $\mathcal{N}_5 = \{2\}$

.

The degree of the subsystem i is defined as

$$d_i = \sum_{j=1}^{N} v_{i,j} \tag{10.4}$$

which denotes the number of directly connected neighbours of subsystem i.

We assume that undirected graph $\mathcal{G}$ is connected. Reference [101] provides comprehensive definition of connected graph. Thus, in order to use the proposed method, we need to look at if the graph $\mathcal{G}$ is connected by evaluating the rank of its Laplacian matrix, $\mathcal{L}$. $\mathcal{G}$ is connected, if and only if rank $(\mathcal{L}(\mathcal{G})) = N - 1$, where $\mathcal{L} = \mathcal{D} - \mathcal{V}$, and $[\mathcal{D}]_{i,i} = d_i$.

10.3 Proposed Solution

In this paper, we aim to solve the steady-state optimal resource allocation problem (10.1) in a decentralized manner without a central coordinator using consensus-based approach and primal decomposition. To this end, we want to propose a framework that so called consensus-based decentralized RTO using primal decomposition as shown in Figure 10.2.

Consider $x_0 \in \mathbb{R}^{m_x}$ as the decision variables for a virtual subsystem 0 to allocate unused available shared resources, problem 1 can be expressed as a problem with equality constraints.

$$\min_{x_0, x_1, \ldots, x_N} \quad f_0(x_0) + \sum_{i=1}^{N} f_i(x_i) \tag{10.5a}$$

$$\text{s.t.} \quad x_0 + \sum_{i=1}^{N} A_i x_i - \bar{\mathbf{x}} = 0 \tag{10.5b}$$

where $f_0(x_0)$ gives constant value. This virtual subsystem can be attached to any physical subsystem.

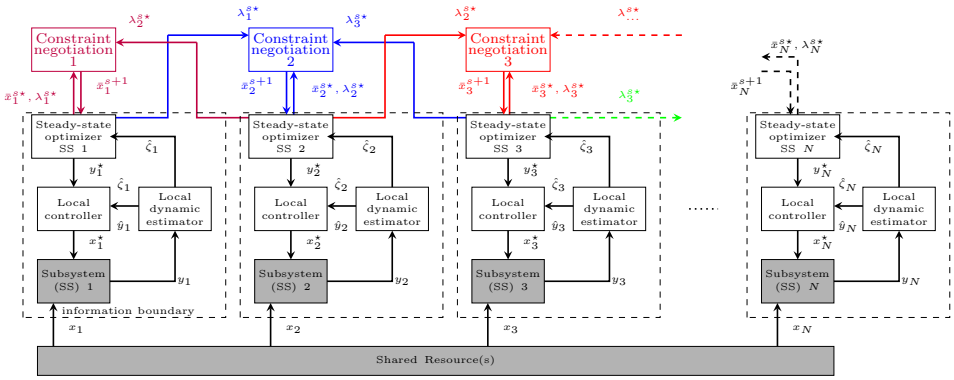


Figure 10.2: The proposed decentralized RTO. Grey box represents the physical system and white box represents the computation block. Purple box and lines represent constraint negotiation process between subsystem 1 and its neighbor ($\mathcal{N}_1$). Blue box and lines represent constraint negotiation process between subsystem 2 and its neighbor ($\mathcal{N}_2$). Red box and lines represent constraint negotiation process between subsystem 3 and $\mathcal{N}_3$. Green lines represents information transmission from subsystem 3 as a neighbor in a constraint negotiation process for other subsystems. As local controllers, MPC needs estimated states, labelled by $\hat{y}_i$, and estimated disturbance, $\hat{\zeta}_i$. For PID as local controllers, $\hat{y}_i = y_i$, which is an outputs measurement, thus dynamic estimator is unnecessary for PID controller but important to estimate disturbance $\hat{\zeta}_i$ for the optimizer.

Defining $\bar{x}_i \in \mathbb{R}^{m_x}$ as shared resource constraints for subsystem i, we can decompose problem (10.1) into $N+1$ different local problems, where $\sum_{i=0}^{N} \bar{x}_i = \bar{\mathbf{x}}$. This decomposition is centralized. To construct decentralized framework, we define $\mathbf{x}_j = \{x_j : j \in \mathcal{N}_i\}$, and $\bar{\mathbf{x}}_{\mathcal{N}_i}$ as available shared resource constraints known by subsystem i and its neighbors $\mathcal{N}_i$. Thus, we can express the neighborhood-wise problem for subsystem i as follows.

$$\min_{x_i, \mathbf{x}_j} \quad f_i(x_i) + \sum_{j \in \mathcal{N}_i} f_j(x_j) \tag{10.6a}$$

$$\text{s.t.} \quad A_i x_i + \sum_{j \in \mathcal{N}_i} A_j x_j - \bar{\mathbf{x}}_{\mathcal{N}_i} = 0 \tag{10.6b}$$

where x_j denotes the decision variables for subsystem j, and $j \in \mathcal{N}_i$.

The Lagrangian of problem (10.6a) is as follows:

$$\mathcal{L}_i(x_i, \mathbf{x}_j, \boldsymbol{\lambda}_{\mathcal{N}_i}) = f_i(x_i) + \sum_{j \in \mathcal{N}_i} f_j(x_j) + \boldsymbol{\lambda}_{\mathcal{N}_i}^{\top} \left(A_i x_i + \sum_{j \in \mathcal{N}_i} A_j x_j - \bar{\mathbf{x}}_{\mathcal{N}_i} \right) \tag{10.7}$$

where $\boldsymbol{\lambda}_{\mathcal{N}_i} \in \mathbb{R}^{m_x}$ is the Lagrange multiplier of the shared resource constraints in the neighborhood i.

As an initial step ($s = 0$), each neighboring subsystem considers all resources consumed as its local shared resource constraints (i.e., $\bar{x}_j^0 = A_j x_j^0$), Thus, consider a primal

decomposition as follows.

$$\min_{x_i} \quad f_i(x_i) \tag{10.8a}$$

$$\text{s.t.} \quad A_i x_i + \sum_{j \in \mathcal{N}_i} \bar{x}_j^s - \bar{\mathbf{x}}_{\mathcal{N}_i}^s = 0 \tag{10.8b}$$

Note that, we now consider constrained problem with equality constraint as we have introduced a virtual subsystem.

Defining $\bar{x}_{\mathcal{N}_i}^s = \bar{\mathbf{x}}_{\mathcal{N}_i}^s - \sum_{j \in \mathcal{N}_i} \bar{x}_j^s$, we can see that problem (10.7) becomes additively separable, where each subproblem is given as a function of the local shadow price $\boldsymbol{\lambda}_i$.

$$\mathcal{P}_i(\boldsymbol{\lambda}_i) := \min_{x_i} \mathcal{L}_i(x_i, \boldsymbol{\lambda}_i) \tag{10.9}$$

where

$$\mathcal{L}_i(x_i, \boldsymbol{\lambda}_i) = f_i(x_i) + \boldsymbol{\lambda}_i^\top \left(A_i x_i - \bar{x}_{\mathcal{N}_i}^s\right) + \frac{\rho}{2} \left\| A_i x_i - \bar{x}_{\mathcal{N}_i}^s \right\|_2^2$$

This is known as augmented Lagrangian decomposition, where $\rho > 0$ is a step length.

Starting from an initial guess $\boldsymbol{\lambda}_i^0$, we solve problem (10.9) and updates the Lagrange multipliers iteratively.

$$x_i^{k+1} := \arg\min_{x_i} \mathcal{L}_i(x_i, \boldsymbol{\lambda}_i^k) \tag{10.10}$$

$$\boldsymbol{\lambda}_i^{k+1} := \boldsymbol{\lambda}_i^k + \rho \left(A_i x_i^{k+1} - \bar{x}_{\mathcal{N}_i}^s\right) \tag{10.11}$$

where x_i^k is optimal resources allocation at iteration k. The Lagrange multipliers $\boldsymbol{\lambda}_i$ denotes the shadow price of the shared resource, which has an economic interpretation of matching the supply and demand of the shared resource from the perspective of subsystem i based on its knowledge within its neighborhood. This is also known as *method of multipliers* for solving problem (10.8). When we combine the decomposability of dual ascent with the method of multipliers, it is also known as alternating direction method of multipliers (ADMM). [64].

When the iteration converges and stop, it means that $A_i x_i^{s\star} = \bar{x}_{\mathcal{N}_i}^s$. If $\boldsymbol{\lambda}_i^{s\star} > 0$, it means that subsystem i needs more resources. When the iteration does not satisfy the constraint and stop, it means that $A_i x_i^{s\star} < \bar{x}_{\mathcal{N}_i}^s$ and $\boldsymbol{\lambda}_i^{s\star}$ indicates that subsystem i can release some of its resource allocation.

Based on these obtained local Lagrangian multipliers, we update local shared constraint for the next step as follows.

$$\bar{x}_i^{s+1} := \bar{x}_i^s + \alpha \sum_{j \in \mathcal{N}_i} \left(\boldsymbol{\lambda}_i^{s\star} - \boldsymbol{\lambda}_j^{s\star}\right) \tag{10.12}$$

where α is a step length.

From the Lagrangian decomposition framework, it can be seen that for the stationary condition of the entire network, we need $\nabla \mathcal{L}(\mathbf{x}^\star, \boldsymbol{\lambda}^\star) = 0$, where every subsystem in the network has the same value of $\boldsymbol{\lambda}^\star$. By updating the local shared constraint at every step as shown in Eq. (10.12), we adjust the local shadow prices $\boldsymbol{\lambda}_i^{s\star}$ ($i \in \mathcal{S}$) in order to reach the common values $\boldsymbol{\lambda}^\star$.

Subsequently, in order to ensure that resource allocation is updated in the entire network, we update $\bar{\mathbf{x}}_{\mathcal{N}_i}$ for the next step as follows.

$$\bar{\mathbf{x}}_{\mathcal{N}_i}^{s+1} = \bar{x}_i^{s+1} + \sum_{j \in \mathcal{N}_i} \bar{x}_j^{s+1} \tag{10.13}$$

Before the entire network converges, $\bar{\mathbf{x}}_{\mathcal{N}_i}^{s+1} \neq \bar{\mathbf{x}}_{\mathcal{N}_i}^{s}$ because $\bar{x}_j^{s+1}$ is updated based on its neighborhood-wise knowledge $\mathcal{N}_j$.

By iteratively executing Eqs. (10.10), (10.11), and (10.12) locally in paralel and (10.13) in neighborhood-wise, we can avoid the need for central coordinator, and we can obtain optimal resource allocation for local RTO in decentralized fashion. Moreover, a subsystem has only limited knowledge about the others subsystems in its neighborhood, and even has no knowledge about others subsystems outside its neighborhood. Thus, a subsystem do not need to share local information such as the detail models, measurements, local constraints, and the objective function, across the different subsystems. The only information that needs to be shared to its neighbor is the local shadow price $\boldsymbol{\lambda}_i^{s\star}$.

10.4 Numerical Example

In this section, we apply the proposed approach on a subsea gas-lifted oil production well network with $N = 5$ subsystems (subsea clusters), operated by different organizations as shown in Figure 10.3. The overall network $\mathcal{G}$ and its neighboring sets ($\mathcal{N}_i$) is given by Figure 10.1 (b). Rank $(\mathcal{L}(\mathcal{G})) = 4$, indicating that $\mathcal{G}$ is a connected graph/network.

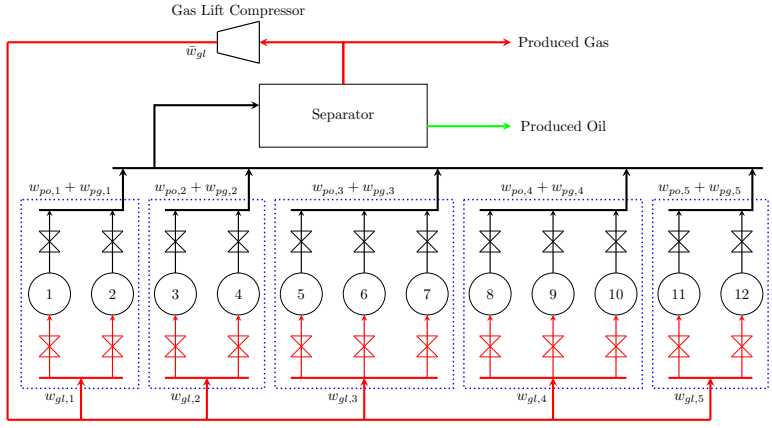


Figure 10.3: A simplified process diagram of a large-scale offshore field with shared gas-lift resource, equipped with constraints in gas-lift compressor.

The produced oil and gas from each subsystem gather in a common topside process facility that has the gas compression station as shown in Figure 10.3. In gas lifted wells, compressed gas is injected into the well tubing in order to reduce the hydrostatic pressure losses and increase oil production. The lift gas supplied from the gas compressor is a common shared resource that must be optimally allocated among the different subsystems.

The objective is to maximize the revenue from the oil production from each subsystems and minimize the costs associated with the gas lift compression. The lift gas w_{gl} is a shared

resource that must be optimally allocated amongst the subsystems. Thus, the optimization problem for problem (10.1) is given as follows.

$$\min_{w_{gl,1},\ldots,w_{gl,N}} \quad -\$_o \sum_{i=1}^{N} w_{po,i} + \$_{gl} \sum_{i=1}^{N} w_{gl,i} \tag{10.14a}$$

$$\text{s.t.} \quad \sum_{i=1}^{N} w_{gl,i} - \bar{w}_{gl} \leq 0 \tag{10.14b}$$

where $\$_o$ is the oil price, and $\$_{gl}$ is the cost of gas compression. Total gas lift constraint is labelled by $\bar{w}_{gl}$. Gas-lift injection rate, $w_{gl,i}$, are the decision variables, $w_{po,i}$ is the oil production rates, which depend on $w_{gl,i}$. The local objective function is given by

$$f_i(w_{gl,i}) = -\$_o w_{po,i}(w_{gl,i}) + \$_{gl} w_{gl,i}$$

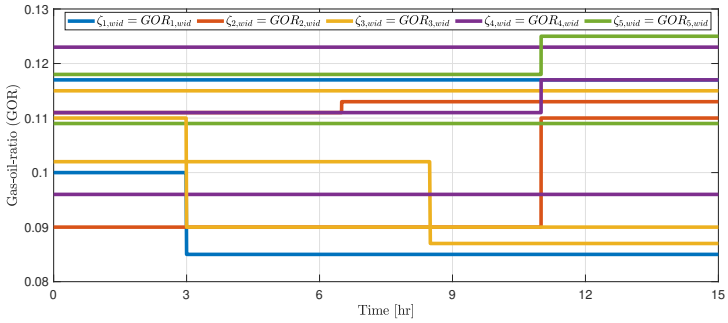


Figure 10.4: GOR variations. labelled by $\zeta_{i,wid}$, of well wid in subsystem i.

Defining $\bar{w}^s_{gl,i}$ as local shared gas lift constraint for subsystem i at step s, the local subproblem can be expressed as follows.

$$w^{s\star}_{gl,i} := \arg\min \quad f_i(w_{gl,i}) \tag{10.15a}$$

$$\text{s.t.} \quad w_{gl,i} - \bar{w}^s_{gl,i} = 0 \tag{10.15b}$$

where $\sum_{i=1}^{N} \bar{w}^s_{gl,i} = \bar{w}_{gl}$.

The gas-to-oil ratio (GOR), which is a reservoir property, is a time varying disturbance for the different wells. In this simulation study, the GOR are assumed to vary as shown in Figure 10.4. High GOR indicates that the well has a lighter fluid requires less amount of gas-lift rate compared to the wells with lower GOR to produce the same amount of oil. GOR is normally local information.

Local RTO problem (10.8) is solved every 60 `sec`. We use extended kalman filter (EKF) as the local dynamic estimator to update the model parameters using transient

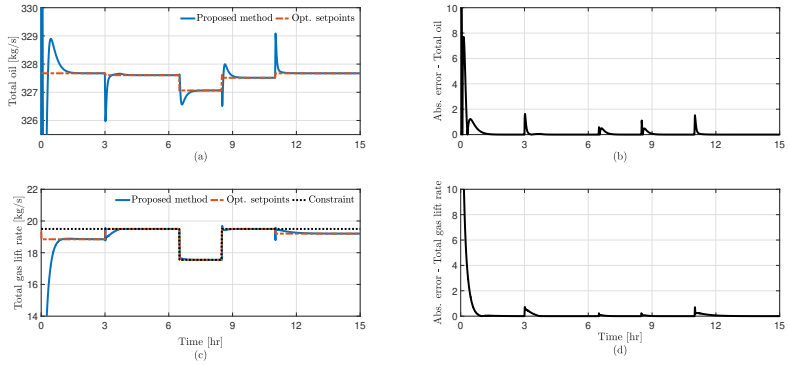


Figure 10.5: Simulation results showing the performance of the proposed framework can reach optimal steady-state setpoints, along with the absolute error between them (total produced oil in top-right subplot, and total gas lift rate in bottom-right subplot).

measurements [10]. Note that the proposed approach is not just restricted to this local RTO approach, and one may instead use any other approach such as dynamic RTO.

In this paper, we use PI controllers for each well for local setpoint, which are tuned using the SIMC tuning rules [52]. The controller is designed with a sampling time of 1 `sec`. Each well uses this controller to drive the wellhead pressure, $w_{ph,i,wid}$, to an optimal setpoint computed by the local optimizer (10.15).

The overall plant is modelled as an Index-1 DAE. The model equations can be found in A. The plant simulator is developed using the `CasADi ver. 3.5.1` toolbox ([53]) in `MATLAB R2019b` and is simulated using the `IDAS` integrator [102]. The simulations are performed on a `2.11 GHz` processor with `16 GB` memory. The simulations are performed for a total simulation time of 15 hours.

We solve the centralized optimization problem (10.14) to obtain ideal optimal setpoints as the baseline to measure the performance of the proposed method.

Figure 10.5(a) shows the simulation results comparing the ideal optimum obtained by solving problem (10.15), and the proposed method. The absolute errors between the ideal optimum and the proposed method are shown in Fig. 10.5(b) and 10.5(d) for total oil production rate and total gas lift injection rate respectively, which indicate that the proposed method is able to converge to the ideal optimum at steady-state.

Figure 10.6(a) shows resource allocation of each subsystem that is updated at every step and changed due disturbances (i.e., GOR and shared resource capacity constraints). In unconstrained case, virtual subsystem store the unused share resource. Figure 10.6(b) shows that local shadow prices converge to certain values at steady-state, which means that each subsystem pays the same price in the steady-state condition. In unconstrained case, no subsystem needs to pay since the shared resource is available. But, when the resource is limited, then they need to pay.

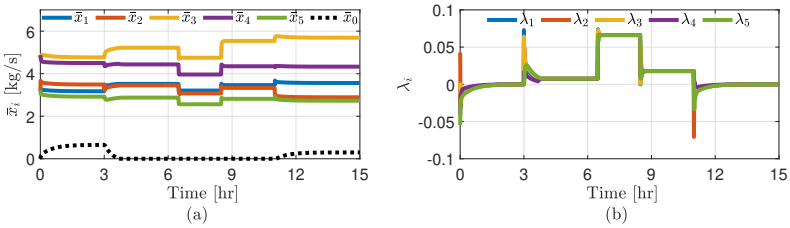


Figure 10.6: The left subplot shows local constraint of each subsystem, including the virtual one. The virtual local constraint is non zero when the constraint is inactive. The right subplot shows local shadow of each subsystem that converges to certain values at steady-state, The shadow prices are zero when the constraint is inactive.

10.5 Discussion

Solving local problems isn not limited to numerical solvers; in favorable matrix conditions, estimating shadow prices can be achieved via an equation detailed in Chapter 8. Moreover, this proposed approach not only eliminates the need for a master coordinator but also enhances information sharing robustness [84], ensuring steady-state optimality even if communication channels fail within a connected graph. Furthermore, an open issue in system-wide optimization is convergence rate, crucial due to limited information sharing across subsystems. Various methods aim to accelerate convergence, such as [103]'s quadratic approximation for coupling constraints and [104]'s gain adaptation.

10.6 Chapter Summary

In this paper, we presented a decentralized RTO framework using primal decomposition for local constraint update for optimal resource allocation with limited information sharing, where we showed that the proposed approach can be an alternative to eliminate the need for central coordinator. The simulation results show that the proposed strategy is able to converge to optimal steady-state setpoint. This framework also allows the negotiation among subsystems within connected neighborhood, and enables system-wide optimal operation. Moreover, this approach has relatively no additional cost, making it attractive for marginal oil production field.

Chapter 11

Self-optimizing Control for Recirculated Gas-Lifted Problem under Limited Measurements

Optimizing subsea oil production systems utilizing recirculated gas lift and limited produced gas treatment capacity presents challenges. Real-time optimization (RTO) is a used method for optimizing such systems, but it is restricted by the lack of reliable sensors and the high cost of developing and updating models. As a result, the RTO is typically executed infrequently, and the optimal set points are not updated in real time, leading to suboptimal plant performance over extended periods. This chapter suggests self-optimizing control (SOC) techniques as an alternative solution that can handle frequent disturbances and drive the plant towards near-optimal performance without requiring frequent model updates or solver use. It compares different SOC structures in recirculated gas-lifted oil production optimization, their advantages, and disadvantages. This chapter is based on the work in [105].

11.1 Introduction

The subsea oil and gas industry has placed greater emphasis on efficient production processes while meeting safety, environmental, and cost-effectiveness requirements. This has led to the development of innovative artificial lift techniques such as gas-lifted subsea oil production optimization to increase the flow rate of oil from a reservoir. This technique involves injecting compressed gas into the wellbore to reduce the hydrostatic pressure of the fluid column, allowing the reservoir to flow more easily. The compressed gas is typically produced gas or gas injected from a separate source. Taking gas from a separate source may be *less commercially attractive* for *offshore* facilities.

Overall, a gas lift system is a forgiving method of enhanced production, in other words, even a poor gas lift design may increase production [59]. However, many optimization studies have been limited by less realistic assumptions, such as a fixed separator pressure and gas lift supply from an separate source [60, 106], which is less likely designed for offshore facilities. Here, we consider a case of *recirculated gas lift oil production system*,

which utilizes produced gas as its total gas lift supply and injects it into wells using a gas lift compressor train. This is the most practically common structure of gas lift oil production system. To improve accuracy, a dynamic separator model that considers varying pressure conditions has also been developed.

When it comes to optimizing a recirculated gas lift subsea oil production system with limited produced gas treatment capacity, one might think that numerical-based real-time optimization (RTO) completed with a dynamic estimator is the obvious solution. This method involves optimizing the process economics using rigorous steady-state models, while disturbances are estimated by the dynamic estimator [10, 11]. However, there are several challenges that this method may not be able to overcome. The challenges are as follows:

1. Costly suitable model development for numerical solver leading to infrequent optimal set point updates: Experienced process engineers may create suitable models, but it can be time-consuming due to numerous parameter updates. Frequent optimization may not be easily performed, and regular set point updates may only occur weekly or even monthly. However, disturbances can occur more frequently, requiring fast time scale self-optimizing control (SOC) structures.
2. Limited available measurement to estimate essential parameters such as disturbances and gradients: Accurate estimations of essential parameters such as disturbance and gradient are necessary for optimization purposes, but they may be hindered by the lack of sufficient or reliable sensors in the measurement. Thus, gradient-based SOC techniques or the use of dynamic estimator may not be preferable, or their application may be limited.
3. Numerical solvers may have issues with numerical robustness: This kind of issue may occur due to several reasons, including not having a solver-friendly model (even though it is a good model for simulation). Furthermore, some solvers may fail to find the solution due to this issue.
4. Costly dynamic model development for dynamic estimator in large and complex systems with multiple units and different timescales can be challenging and time-consuming: This dynamic model is required for dynamic estimator, and without a good dynamic model, parameter estimation may not be accurate or may provide non-sense estimation.

Hence, alternative solutions, such as SOC, are considered to address these challenges. This work explores several potential self-optimizing control structures for recirculated gas-lifted subsea oil production systems.

11.2 Problem Description

This section describes a case study designed to explore the possibility of identifying the most effective and suitable self-optimizing control structure that can maximize profits, even under conditions of reservoir uncertainty and limited produced gas treatment capacity.

11.2.1 Recirculated Gas Lifted Subsea Oil Production System

The recirculated gas lift oil production system is depicted in Figure 11.1. The system consists of six (6) gas lifted-oil producing wells, riser, separator, gas lift compressor train

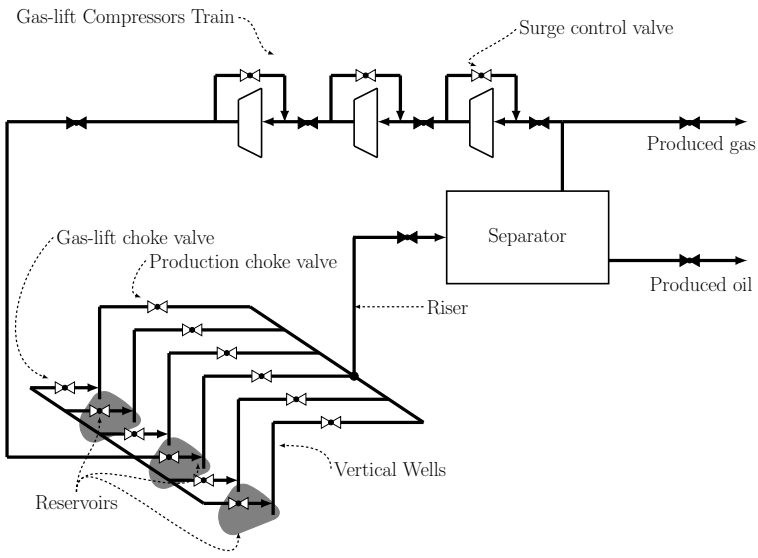


Figure 11.1: The recirculated gas lift oil production system. Black valves represent pressure drop. The opening of the white valves can be manipulated in practice.

(series of three centrifugal compressor), and gas lift supply line. The model is based on extensive research, modification and integration from various sources, including [107, 60, 108, 109, 110], to name a few. The model equation is available in Appendix A. By incorporating more realistic assumptions, we aim to provide a more realistic representation of the gas lift oil production process, which has not been previously discussed to the best of the author's knowledge.

11.2.2 Steady-state Optimization Problem Formulation

The problem at hand involves determining the optimal gas lift choke valve (GLC) positions and surge control valve (SCV) positions for each gas lift compressor to maximize revenue from produced oil, minimize energy consumption of the compressor train, and satisfy operational constraints. The production choke valves (PCV) are assumed to be fixed and fully open (to hold convex problem assumption), regardless of the disturbance being considered. Additionally, fixed angular velocity is assumed for each gas lift compressor for practical reasons. The steady-state optimization problem formulation can be expressed as follows:

$$\min_{\mathbf{u}} \quad J(\mathbf{u}, d) = -p_{oil} w_{os} + p_{en} \Phi_{gl} \tag{11.1a}$$

$$\text{s.t.} \quad g_{z_{gl,i}}(\mathbf{u}, d) : z_{gl,i} - 1 \leq 0 \quad i = 1, \ldots, 6, \tag{11.1b}$$

$$g_{z_{s,i}}(\mathbf{u}, d) : -z_{s,i} + 0 \leq 0 \quad i = 1, \ldots, 3, \tag{11.1c}$$

$$g_{s_i}(\mathbf{u}, d) : s_i - \bar{s}_i \leq 0 \quad i = 1, \ldots, 3, \tag{11.1d}$$

$$g(\mathbf{u}, d) : w_{gs} - \bar{w}_{gs} \leq 0 \tag{11.1e}$$

Here, the manipulated variables (MVs) are as follows:

$$\mathbf{u} = \begin{bmatrix} z_1 & \dots & z_{n_\mathbf{u}} \end{bmatrix}^\top = \begin{bmatrix} z_{gl,1} & \dots & z_{gl,6} & z_{s,1} & \dots & z_{s,3} \end{bmatrix}^\top$$

where $z_{gl,i}$ is the position of GLC of well i, $z_{s,i}$ is the position of SCV of gas lift compressor i, and $n_\mathbf{u} = 9$ is the number of MVs. The disturbance d comes from reservoir uncertainty. The produced oil rate, produced gas rate, and maximum produced gas treatment capacity are represented by w_{os}, w_{gs}, and $\bar{w}_{gs}$, respectively. The price of oil and energy are represented by p_{oil} and p_{en}, respectively.The energy consumed by the gas lift compressor train is represented by, $\Phi_{gl} = \sum_{i=1}^{3} \Phi_{gl,i}$. The surge of gas lift compressor i is represented as s_i, and the associated limit is represented as $\bar{s}_i$. The input constraints for the GLC and SCV are shown in constraints (11.1b) and (11.1c), respectively. The surge constraint of gas lift compressor i is shown in constraint (11.1d), and the produced gas constraint is shown in constraint (11.1e).

In addition to constrained variables, we assume that the available measurements are:

$$\mathbf{y} = \begin{bmatrix} p_{bh,2} & p_{wh,2} & p_{d,3} & p_s \end{bmatrix}^\top$$

Here, $p_{d,3}$ represents the discharge pressure of the gas lift compressor train, and p_s represents the separator pressure. Both of them are the *artificial boundaries* in the previous studies. Meanwhile $p_{bh,2}$ and $p_{wh,2}$ are bottom hole and wellhead pressure of well 2, respectively.

11.2.3 The Nominal Optimal Operating Point

When the numerical solver finds the optimal solution by solving problem (11.1), a nominal optimal steady-state operating point can be obtained. This process most likely occur at a slow and infrequent time scale. The obtained decision/MVs associated with this nominal optimal operating point are as follows.

$$\mathbf{u}^\star = \begin{bmatrix} 0.88 & 0.49 & 1.00 & 0.64 & 0.60 & 0.81 & 0.00 & 0.00 & 0.00 \end{bmatrix}^\top \tag{11.2}$$

Further, the active constraints associated with this nominal optimal operating point are $g^\star_{z_{gl,3}}$, $g^\star_{z_{s,1}}$, $g^\star_{z_{s,2}}$, $g^\star_{z_{s,3}}$, $g^\star_{s,1}$, $g^\star_{s,2}$, $g^\star_{s,3}$, and $g^\star$. It means that with a given disturbance d, optimal operation happen if constraints (11.1c)-(11.1e) are active, and the position of GLC 3 is fully open.

11.2.4 Reservoir Uncertainty (Disturbances)

The uncertainty in the reservoir, denoted by d, is related to the gas-oil-ratio of a particular well that exhibits unstable conditions. To be specific, we assume that the disturbance is originating from well 2, and is represented by $d = GOR_2$.

Multiphase flow meter (MPFM) has recently been recommended for frequent estimation of the GOR or other parameters of a well, replacing the use of test separator. Despite its promising performance, only 3% of the 70,000 active oil producing wells worldwide have utilized MPFM [111]. Any new report will most likely provide larger percentage of MPFM utilization as this technology is relatively recent, and there are many old wells are still producing. To address the majority of the case, we assume that we are unable to frequently and accurately estimate the GOR of the well and rely on the historical data of

well testing using test separator to obtain the GOR fluctuation range as the information we have. Therefore, in this case, we assume that based on well testing, this disturbance to fluctuate by up to $\pm 2.5\%$. If the disturbance increases by 2.5%, the optimal state is achieved when constraint (11.1e) is active. On the other hand, the optimal state is attained when constraint (11.1e) is inactive.

11.3 Self-optimizing Control Structure

This section provides a brief introduction to self-optimizing control, and its implementation on the problem described in Section 2. For a more detailed review of SOC, please refer to the survey paper by [33].

11.3.1 General Principle

The goal of self-optimizing control (SOC) is to achieve near-optimal operation by controlling a combination of selected variables $\mathbf{c} \in \mathbb{R}^{n_\mathbf{u}}$ at their constant set points [8]. The objective is to maintain these variables at their set-points despite the presence of varying disturbances, resulting in an acceptably low loss during operation. This is accomplished by utilizing feedback from the appropriate combination of measurements to counteract the effect of unmeasured disturbances d. It is important to note that this technique *assumes the same active constraints remain active for all values of the disturbances, and these constraints are controlled*.

This goal of this technique can be achieved through a heuristic method, null space method introduced by [29], or a combination of them. In the following sections, several self-optimizing control structures applicable to a recirculated gas-lifted subsea oil production well are described. Constraints related to limited produced gas treatment capacity are considered.

11.3.2 Active Constraint Control

When the optimal set points remain unchanged, the position of the GLCs and the SCVs are typically maintained, which implies that the MVs remain constant at the values specified in Eq. 11.2. This configuration is referred to as *Structure 1*.

In this study, both the produced gas and surge line constraints are active at the nominal optimal operating point. To achieve effective control, the MVs (valve positions) should be paired with the constrained variables closely. This implies that the relationship between the constrained variables and MVs should have a high gain for better control, but the MVs' initial nominal optimal values should not readily lead to saturation in controlling the constraint. Therefore, manipulating GLC 3 to control the produced gas constraint (11.1e) is not recommended. Instead, we suggest using $z_{gl,5}$ (GLC 5) to control constraint (11.1e) as it has enough maneuvering room to handle the disturbance variation caused by GOR_2 increasing or decreasing by up to 2.5%. GLC 5 also has the highest gain compared to the remaining GLCs. This ensure effective control of active constraints which is essential, according to [8]. This relationship can be expressed as

$$z_{gl,5} \rightarrow g(\mathbf{u}, d)$$

With respect to the gas-lift compressors, the most efficient operating point is achieved when the surge constraints (11.1d) are active. Although the SCVs are already saturated (fully closed) at the nominal optimal, in some cases, opening the SCV is necessary to

ensure that the surge line constraint remains active. In other cases, when the SCV is fully closed (and already saturated), the discharge pressure of the gas lift compressor train is automatically adjusted, and the discharge pressure is considered floating and will never be saturated. As a result, we propose pairing the SCV with the associated surge line constraint (11.1d), which can be expressed mathematically as

$$z_{s,i} \rightarrow g_{z_{s,i}}(\mathbf{u}, d)$$

for $i = 1, 2, 3$. By incorporating these active constraint controllers, we refer to this configuration as *Structure 2*.

11.3.3 Heuristic Method

The gas production of a reservoir can fluctuate due to various factors, such as the gas-oil-ratio (GOR). When well 2 produces more gas from the reservoir, as indicated by the increase in parameter GOR_2, the active constraint controller adjusts the position of GLC 5 for maximum produced gas handling capacity. This ensures no steady-state violation on the produced gas constraint (11.1e). However, if the amount of gas produced by well 2 decreases, the calculated position of GLC 5 from the active constraint controller may become unsuitable, *rendering the assumption of self-optimizing control invalid*. In such cases, it is necessary to identify an alternative self-optimizing controlled variable that is good and measurable for GLC 5. The position of GLC 5 itself is an obvious option. Additionally, we assume that the active surge constraint (11.1d) is the best choice for all values of expected disturbance.

When selecting a self-optimizing controlled variable, it is also important to consider its proximity to the source of the disturbance, which, in this case, is GOR_2. One option for stabilization is to use a readily available and measured variable, such as the bottom hole pressure of well 2. Thus, the position of GLC 2 can be replaced with the bottom hole pressure of well 2 ($z_{gl,2} \rightarrow p_{bh,2}$). However, the bottom hole pressure may not always be a practical variable to measure, as damage to the associated sensor may require costly replacements due to its location. As an alternative, we can use wellhead pressure ($z_{gl,2} \rightarrow p_{wh,2}$), which is easier to maintain and replace, although it may not be the closest variable to the source of the disturbance.

Another possible self-optimizing controlled variable is the differential pressure between wellhead and bottom hole pressure, which can be used to maintain the flow rate from the well by keeping the differential pressure constant (This strategy will be affected by density changes and rates/friction). This can be performed by embedding an observer that considers all necessary pressure profile in the well to establish analytical redundancy if the sensor in the bottomhole underperforms.

Based on these heuristic steps, we construct three potentially promising control structures, named *Structure 3, 4, and 5*, as illustrated in Figure 11.2. However, this heuristic method does not explore other measurement sets and remaining MVs, which may be impractical and time-consuming. This is the limitation of heuristic method.

11.3.4 Null Space Method

Combining different measurements can result in a better performing control structure. To incorporate unexplored measurements and MVs in a systematic way, we use the null space method. Let us assume that we have $n_{\mathbf{u}}$ independent unconstrained free variables

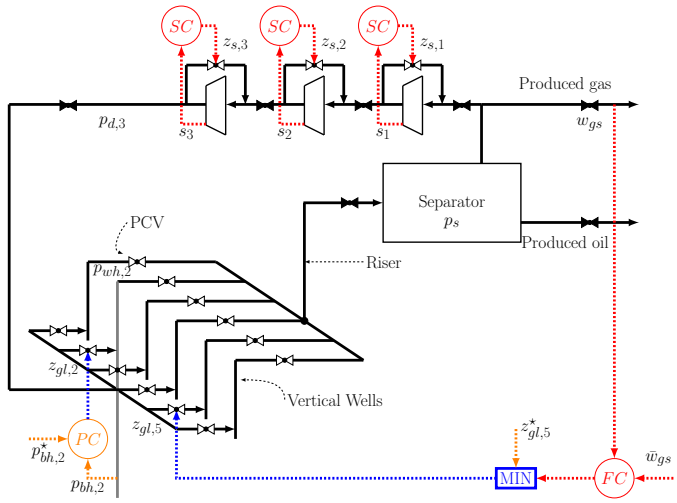


Figure 11.2: *Structure 1* removes all controllers and selector, while *Structure 2* removes the pressure controller and selector, leaving only the online produced gas flow controller and surge controllers. *Structure 3* considers a `MIN` selector that allows the active constraint to change with $z_{gl,5}$ as the self-optimizing controlled variable when GOR_2 decreases, and $p_{bh,2}$ as another self-optimizing controlled variable for stabilization purposes. *Structure 4* considers wellhead pressure of well 2 ($p_{wh,2}$) instead of $p_{bh,2}$, and *Structure 5* considers the differential pressure of well 2 ($\Delta p_{bw,2} = p_{bh,2} - p_{wh,2}$) instead of $p_{bh,2}$. Black valves represent pressure drops.

$\mathbf{u}$, $n_\mathbf{d}$ independent disturbances $\mathbf{d}$, and $n_\mathbf{y}$ independent measurements $\mathbf{y}$. Our goal is to obtain $n_\mathbf{c} = n_\mathbf{u}$ independent controlled variables $\mathbf{c}$ that are linear combinations of the measurements, which can be expressed as:

$$\mathbf{c} = \mathbf{H}\mathbf{y}$$

To achieve this, we use the optimal sensitivity matrix

$$\mathbf{F} = \frac{\partial \mathbf{y}^\star}{\partial \mathbf{d}}$$

which is evaluated with constant active constraints. If $n_\mathbf{y} \geq n_\mathbf{u} + n_\mathbf{d}$, we can select the matrix $\mathbf{H}$ in the left null space of $\mathbf{F}$, such that $\mathbf{HF} = \mathbf{0}$. This choice of $\mathbf{H}$ ensures that fixing $\mathbf{c}$ at its nominal optimal value is first-order optimal for disturbances $\mathbf{d}$, resulting in zero loss as long as the sensitivity matrix $\mathbf{F}$ remains unchanged [29]. However, to prevent unnecessary complexity and cost, the number of measurements used in the structure should be limited. Ideally, a cost-benefit analysis should be performed to determine the optimal instrumentation for the plant.

Furthermore, null space method still assumes that the same active constraints remain active for all values of the disturbances. To *relax* this assumption into the design, we use a selector for active constraint switching. Similar to the heuristic method, we choose $z_{gl,5}$ as the self-optimizing controlled variable when GOR_2 decreases, using the `MIN` selector. This strategy eliminates the need to explicitly estimate GOR_2. Instead, any variation in

GOR_2 is indirectly detected through changes in the total produced gas w_{gs}, which serves as the CV. The selector is determined by the relationship between MV and CV, as described by [51]. Moreover, when the well is equipped with an adequate measurement set, it becomes feasible to estimate GOR_2 using a straightforward model or data processing techniques.

Ideally, we can implement null space method for unconstrained case, and construct another control structure for this case. This control structure can also be switched using selector [33]. However, the optimal operating point of the other case is normally unknown in practice. Therefore, an insight from previous heuristic method is necessary.

Utilizing single MV (Structure 6):

In addition to *flexible* active constraint controls (that is combined with `MIN` selector), this structure (denoted as *Structure 6*) consider the following measurement set: $\mathbf{y} = \begin{bmatrix} p_{bh,2} & p_{wh,2} \end{bmatrix}^\top$.

It is possible to replace one of the measurements with the position of GLC 2. However, doing so would result in a loss of control over either the bottom hole or wellhead pressure, which is undesirable as maintaining proper stabilizing control is crucial. Here, $z_{gl,2}$ is considered as an MV for stabilizing control.

To obtain the sensitivity matrix $\mathbf{F}$, numerical methods were employed. The null space method was then utilized to determine the optimal matrix $\mathbf{H}$, which corresponds to the following controlled variables:

$$c = 0.521 p_{bh,2} + 0.854 p_{wh,2} \tag{11.3}$$

This structure has the same measurement elements as *Structure 3-5*. Hence, comparing them is justified.

Based on branch and bound algorithm [112] and the requirement to keep at least either $p_{bh,2}$ or $p_{wh,2}$ as one of the measurements, we found that *Structure 6* is still the best option.

Utilizing two MVs (Structure 7):

As we consider two MVs (i.e., $z_{gl,2}$ and $z_{gl,4}$) and one disturbance ($n_{\mathbf{d}} = 1$), three measurement is required ($n_{\mathbf{y}} = 3$). Further, as we have four measurement candidate and we want to include at least either $p_{bh,2}$ or $p_{wh,2}$ in the measurement set, four combinations of measurements need to be evaluated. Unfortunately, the branch and bound algorithm [112] is not applicable because $z_{gl,4}$ may be saturated when GOR_2 increases up to 2.5%. Therefore, the only possible evaluation technique to determine the best combination is the average steady-state lost. Based on this technique, we found the best combination of measurement is: $\mathbf{y} = \begin{bmatrix} p_{bh,2} & p_{wh,2} & p_{d,3} \end{bmatrix}^\top$, and the corresponded controlled variables are:

$$\mathbf{c} = \begin{bmatrix} \mathbf{c}(1) \\ \mathbf{c}(2) \end{bmatrix} = \begin{bmatrix} 0.520 p_{bh,2} + 0.854 p_{wh,2} - 0.012 p_{d,3} \\ 0.041 p_{bh,2} - 0.012 p_{wh,2} + 0.999 p_{d,3} \end{bmatrix} \tag{11.4}$$

Based on Relative-Gain-Array (RGA) analysis [46], the recommended pairing is: $z_{gl,2} \rightarrow \mathbf{c}(1)$, and $z_{gl,4} \rightarrow \mathbf{c}(2)$.

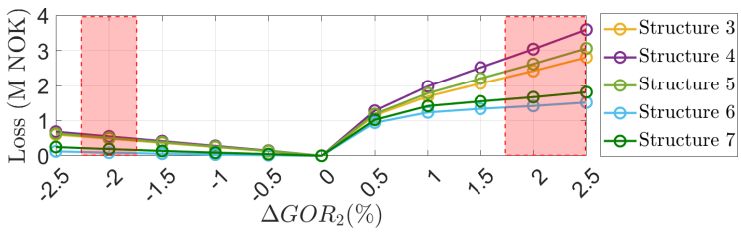


Figure 11.3: Comparison of steady-state monthly loss of different structures. Red boxes indicates estimated loss evaluation.

11.4 Results and discussions

11.4.1 Estimated Loss Evaluation

One of the motivation of utilizing SOC is numerical robustness issues. In this simulation, we evaluate problem (11.1) with eleven (11) different value of GOR_2 ranging from 97.5% to 102.5% of GOR_2 nominal with 0.5% interval. The plant simulator is developed using the `CasADi ver. 3.5.1` toolbox ([53]) in `MATLAB R2019b` and is simulated using the `IDAS` integrator. The simulations are performed on a `2.11 GHz` processor with `16 GB` memory. The solver used is `IPOPT` with `MUMPS` as linear solver. Note that there are many linear solvers available and each has its own numerical limitation [113]. Table 11.1 shows solver's performance, indicating that the solver is not always possible to obtain optimal solutions. For those cases, we estimate the optimal solutions based on the linear regression from the closest available two solution points. Thus, we consider *an optimal profit set* containing the solver-based optimal profit and the estimated optimal profit.

Steady-state loss is the difference between the profit obtained by the discussed/proposed control structure and the optimal profit from *the optimal profit set*. Table 11.2 summarizes steady-state loss obtained by different control structures when GOR_2 increases or decreases up to 2.5%. As described in previous sections, structure 1 keeps the position of GLCs and SCVs unchanged, while structure 2 implements active constraint control on both surge constraints and export/produced gas constraint (see section 11.3.2). In addition to active constraint control, structure 3, 4, and 5 introduces bottom hole, wellhead, and the differential pressure as self-optimizing controlled variable, respectively (see section 11.3.3). Finally, structure 6 and 7 utilize a combination of 2 and 3 measurements as self-optimizing controlled variable, respectively (see section 11.3.4).

Table 11.1: Solver's success and fail performance

-2.5%	-2.0%	-1.5%	-1.0%	-0.5%
✓	✗	✓	✓	✓
0.5%	1.0%	1.5%	2.0%	2.5%
✓	✓	✓	✗	✗

11.4.2 Active Constraint Control

Structure 1 lacks a control strategy to handle reservoir uncertainty. This option is not ideal because an increase in GOR_2 leads to an increase in the amount of total produced gas. In steady-state, this violates the maximum gas handling capacity constraint. By installing constraint controllers, described in *Structure 2*, these violations are successfully eliminated. However, it is worth noting that the expected steady-state loss when GOR_2 decreases is much higher compared to *Structure 1* (see Table 11.2).

11.4.3 Heuristic Method

When GOR_2 decreases, in *Structure 3-5*, the self-optimizing controlled variable for GLC 5 is the position of the GLC 5, while the SCVs handle the surge constraints. As a result, *Structure 3-5* has successfully decreased the expected average steady-state loss compared to *Structure 2* (see Table. 11.2). In general, *Structure 3* outperforms other heuristic-based structures.

Depending on the reservoir dynamics and the possible controlling tuning, each control structure may reach the steady-state with different time steps/time constant. In addition to that, the selection of control structure should also consider the specific situation and constraints of the field. If immediate stabilization action is required and there is a reliable bottomhole sensor, *Structure 3* would be a good option. On the other hand, if maintaining the wellhead pressure is more feasible than installing and maintaining a bottomhole sensor, *Structure 4* may be preferable. *Structure 5* requires both a bottomhole and wellhead pressure sensor and is only viable if both sensors are functional.

The heuristic method is an intuitively useful tool for engineers. For instance, providing alternative self-optimizing controlled variable for GLC 5 when GOR_2 decreases is determined heuristically. This is significant considering null space method assumes the same active constraint for any disturbance values. While this method may provide valuable insights, it can also require significant effort and resources to achieve meaningful results. Therefore, it is essential to carefully consider the benefits and drawbacks of using the heuristic method in each specific situation, taking into account factors such as the scope of the problem, available resources, and the desired outcome.

Table 11.2: Steady-state monthly loss

Control Structure	$-2.5\%\ GOR_2$	$+2.5\%\ GOR_2$ (est.)
1	NOK 59.544	Inf
2	NOK 6.116.745	NOK $\sim$ 3.444.831
3	NOK 604.897	NOK $\sim$ 2.810.376
4	NOK 686.095	NOK $\sim$ 3.595.481
5	NOK 633.027	NOK $\sim$ 3.065.285
6	NOK 124.246	NOK $\sim$ 1.523.036
7	NOK 248.667	NOK $\sim$ 1.817.930

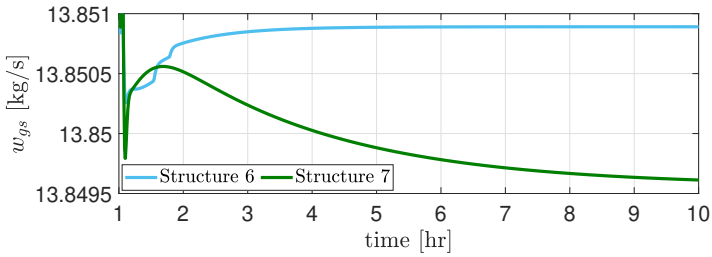


Figure 11.4: Comparison of transient performance of *Structure 6* and *Structure 7* (consider $p_{d,3}$ in the measurement combination)

11.4.4 Null Space Method

As expected, with the same element of measurement set, *Structure 6* outperforms control *Structure 3-5*, developed heuristically (see Figure 11.3). When we consider more candidate of measurement, and utilize branch and bound algorithm to find the optimal measurement set, we found that *Structure 6* still outperforms other possible control structures, containing *artificial boundaries*, i.e, separator pressure and discharge compressor pressure. This is an essential result to demonstrate the importance of eliminating *artificial boundaries*.

Increasing the number of the use of measurement set ($n_{\mathbf{y}} = 3$ in *Structure 7*) may not always leading to lower steady-state loss than *Structure 6*, for instance, if the additional used MV is saturated (see Figure 11.3). In addition, null space method may maximize the use of available MVs by providing more measurement but it does not take into account the effect of the location of the suggested measurement element. For instance, *Structure 7* has discharge pressure in the measurement set which is located far away from its MVs (GLCs). As a consequence, it consumes more time in reaching the steady-state, as shown in Figure 11.4. This comparison obtained with controller tuned using Simple IMC rule [52].

11.4.5 Further Discussion Points

The wells interact with each other, creating back pressure in the process. This means that disturbing one well affects others due to this interaction. In Chapter 4, we address this issue by proposing a problem reformulation and introducing the concept of solution predictors. However, in this chapter, we regard it as a natural behavior of a well network.

In this study, we also assume that we have undertaken relatively robust data pre-processing/pre-conditioning steps to minimize noise or erroneous data until reaching an acceptable composition. Without adequate data pre-processing, the controller may respond more to noise rather than the disturbance, thus impeding the achievement of the self-optimizing control structure's objective.

In practical scenarios, actuators often have several limitations. For example, GLCs are frequently not continuous but rather operate in a stepwise manner. This design choice inevitably affects the performance of the optimizing-control structure. Consequently, the convergence rate may decrease, and consistent ripples may emerge. Economically, this can be unfavorable or lead to divergence if the step size is too large.

Another noteworthy constraint in this case study is related to water treatment capacity. Assessing this constraint necessitates including water-cut in the model. When employing a self-optimizing control strategy, it may be necessary to allocate an available GLC to manage the water treatment capacity constraint, potentially reducing oil production to meet this requirement.

In practice, additional constraints may arise. At times, determining self-optimizing controlled variables may not even be feasible due to a lack of available manipulated variables, leading to what is typically termed a fully constrained case. Ultimately, addressing this issue becomes a matter of tailoring solutions to the specific case at hand.

11.5 Conclusion

This paper investigated the possibilities of applying self-optimizing control to a recirculated gas lifted subsea oil well production optimization. This study reconfirms the previous work that self-optimizing control can be an alternative for optimization strategy without solver.

It was found that the most recommended control structure is *Structure 6*. This structure uses null space method in determining the optimal combination of controlled variables, uses a selector to allow active constraint region switching, and consider a required stabilizing control. This concludes that both heuristic and null space method are necessary and comply one to another.

As future works, we suggest considering cascade controller to solve the issue of having saturated MVs. In addition, it is also interesting design a control structure for a more realistic case where the surge constraint may be inactive, measurable GOR (including using embedded observer), multiple well with unstable GOR, and PCVs as manipulated variables.

Closing Remarks

Chapter 12

Concluding Remarks and Future Works

12.1 Concluding Remarks

This main objective of this thesis is to suggest optimization strategies that have higher chances for practical implementation for a complex and/or large-scale oil production operation system that consists sub-processes with varying timescales, and potentially numerous constraints. Considering the common working culture of this industry, this thesis recommend to utilize the common, well-known and established tools such as PID controllers and selectors, with occasional use of small-scale numerical solvers for specific cases. Furthermore, experimental works were also conducted to validate the proposed optimization strategies.

To address this goal, this thesis started by examining a small-scale process system with many constraints. Due to frequent disturbances, the active constraint configuration may change for optimal operation. The most common existing approach to handle this active constraint switching is reduced-gradient approach. However, this approach is becoming impractical for a system that has too many constraints. As a solution, this thesis suggested primal-dual framework where active constraint switching can be handled with a fixed control structure.

While the Primal-dual framework adeptly manages active constraint switching, its drawbacks include limited control over tight constraints and heavy reliance on accurate real-time gradient estimations. These limitations are particularly problematic for critical constraints in the oil and gas sector, where safety cannot be compromised. With stricter environmental regulations, failure to tightly control constraints could have substantial economic and reputational impacts. To mitigate these issues, this thesis proposed integrating an override constraint controller and introducing auxiliary constraints within the primal-dual framework.

The primal-dual framework also offers another powerful benefit: system decomposition into manageable subsystems. This enables the breakdown of large-scale oil production systems into smaller and more manageable units. Known as dual-based distributed

feedback-optimizing systems, this framework facilitates the coordination of subsystems for system-wide optimal performance. However, not all systems are easily decomposable due to interactive subsystems. To address this, this thesis proposed dual-based distributed feedback-optimizing control with solution predictors that introduce intermediate solution variables estimated via feedback mechanisms. These variables allow the decomposition of interactive large-scale systems. Experimentally validating this approach, the thesis compared it with numerical-solver based RTO like Hybrid RTO for decomposable optimization problems.

Similar to primal-dual, dual-based distributed feedback optimization relies on accurate real-time gradient estimations but offers clearer insight into the effect of non-performing local gradient estimations. This facilitates systematic pairing for an override controller to minimize economic loss in the presence of such issues. Therefore, this thesis suggested a systematic pairing procedure to minimize the economic loss.

Given its need for tight constraint control, override requests often entail aggressive actions but are practically restricted by input rate limits. For instance, control valve actuators may only adjust in limited step size. Moreover, multiple manipulated variables often collaborate to manage a critical constraint; for instance, gas-lift valves from various wells might collectively regulate a shared export gas constraint. Thus, this thesis proposed several possible procedures to facilitate multi-input overrides.

While dual-based distributed feedback-optimizing system offers advantages, it may lose profit during transient in handling of a total input constraint, even with override. This thesis proposed a primal-based distributed feedback-optimizing system that is capable of handling a total input or an output constraint. Experimental validation demonstrates the benefit of primal-based feedback-optimizing system in managing a total input constraint.

Both primal-and dual-based distributed feedback-optimizing systems efficiently coordinate large-scale processes for system-wide optimal performance. However, their reliance on a central coordinator poses challenges, particularly in decentralized contexts like marginal offshore oil fields with multiple company interests. This case may lead to impartiality concerns and heightened technical risks due to increase information exchange (each subsystems has a communication channel to central coordinator). To mitigate these issues, this thesis proposes a graph-based primal-based distributed feedback-optimizing system where information sharing is limited to neighboring subsystems and ensuring robustness in achieving optimal performance even with acceptable communication channel failures.

Addressing the lack of reliable sensors in recirculated gas-lift systems, this thesis extended the existing gas-lift model to a more realistic one by incorporating separator and compressor models, while also integrating gas-lift recirculation into the well. It proposed utilizing self-optimizing control to achieve near-optimal performance, addressing the limitation of standard RTO approaches due to sensor reliability issues.

In summary, no single approach can address all problems. Yet, the ability to break down problems into subproblems is crucial. This allows for tailored approaches, selected based on the specific nature of each subproblem, as outlined in Table 12.1.

Table 12.1: Summary of Approaches Discussed in This Thesis

	(a)	(b)	(c)	(d)	(e)	(f)	(g)	(h)	(i)	(j)
S-I	Y	N	Y	N	Y	N	Y	N	N	Y
S-II	N	Y	N	Y	Y	Y	N	Y	Y	Y
S-III	Y	Y	N	Y	N	N	Y	N	N	N
S-IV	Y	Y	Y	N	N	N	Y	N	N	Y
S-V	Y	Y	N	Y	Y	Y	N	Y	N	N
S-VI	Y	Y	Y	N	Y	Y	N	Y	N	N
S-VII	Y	Y	Y	Y	N	N	Y	N	N	Y
S-VIII	Y	Y	Y	Y	Y	Y	N	Y	N	N
S-IX	Y	Y	Y	N	Y	Y	Y	Y	N	N
S-X	Y	Y	Y	Y	Y	Y	Y	Y	N	N
S-XI	Y	Y	Y	Y	Y	Y	Y	Y	N	N

"Y": the approach has satisfied the success parameter
"N": the approach does not satisfied the success parameter

where the approaches are:

- S-I: Steady-state RTO with dynamic model adaptation (also known as Hybrid RTO)
- S-II: Traditional Self-optimizing Control
- S-III: Reduced gradient hybrid RTO
- S-IV: Dual-based hybrid RTO
- S-V: Reduced gradient FOC
- S-VI: Primal dual FOC
- S-VII : Dual-based hybrid RTO with override
- S-VIII : Primal dual FOC with override
- S-IX : Dual-based Feedback-optimizing system
- S-X : Dual-based Feedback-optimizing system with override
- S-XI : Primal-based Feedback-optimizing system

and the parameters we assess and compare are:

- (a): Is steady-state optimal achieved?
- (b): Is constraint controlled *transparently* (see Definition 2.1)?
- (c): Is flexible for active constraint switching?
- (d): Is constraint controlled *directly* (on fast time scale)?
- (e): Is applicable for less than twice differentiable Lagrange function?
- (f): Is numerical solver avoidable (computation time)?
- (g): Is it recommended for complex and large system?
- (h): Is input filter (or additional setpoint controller) avoidable?
- (i): Is parameters and states dynamic estimator avoidable?
- (j): Is gradient estimator avoidable?

12.2 Recommended Future Works

Based on the conducted research works, the following research directions can extend the work presented in this thesis:

- A standard dual-based feedback-optimizing system is theoretically struggle with serial interconnection subsystems due to assumed acceptable constraint violation, while the coupling constraint linked to artificial boundaries presents a hard constraint. This thesis suggested three approaches: dual-based feedback-optimizing systems with override or solution predictors, and primal-based feedback-optimizing systems with compensators to address constraint associated with artificial boundary. Over the past four years, the author has been focusing on implementing these approaches in a serial connection scenario involving multiple parallel production wells, a separation system, gas export compressors, oil pumps, and gas lift compressors. However, this work remains ongoing at the time of thesis submission, making it intriguing to observe and analyze the results in such a practical setting.

- As discussed in Chapters 8 and 9, a key challenge in primal-based distributed feedback-optimizing systems arises from assuming a unique solution for estimating local Lagrange multipliers. This becomes particularly problematic when numerous coupling constraints are present, leading to a higher likelihood of non-uniqueness solution. Thus, exploring solutions to this issue becomes imperative. Addressing this challenge could potentially disrupt the conclusion of the generality of dual-based feedback-optimizing systems.

- As indicated in Table 12.1, the most unresolved question revolves around the necessity of parameters, states, and gradient estimators. While a parameters and states estimator might result in inaccurate estimations due to structural mismatches in the plant model, focusing on accurate real-time gradient estimation could circumvent this issue. Extremum seeking control offers one possibility but tends to converge slowly and may struggle with multiple gradient estimations in interactive systems. Alternatively, real-time data-based techniques, such as least square error or machine learning tools, offer promising avenues for gradient estimation. Noteworthy is the modifier adaptation with quadratic approximation discussed in [114], serving as a potential starting point for real-time data-based gradient estimation.

- The last but not the least, it is crucial to consistently test and validate coordinated feedback-optimizing systems across diverse practical cases and using the findings to develop new features.

Bibliography

[1] L.A. Saputelli, S. Mochizuki, L. Hutchins, R. Cramer, M.B. Anderson, J.B. Mueller, A. Escorcia, A.L. Harms, C.D. Sisk, S. Pennebaker, J.T. Han, A. Brown, C.S. Kabir, R.D. Reese, G.J. Nunez, K.M. Landgren, C.J. McKie, and C. Airlie. Promoting Real-Time Optimization of Hydrocarbon Producing Systems. In *SPE Offshore Europe Conference and Exhibition*, pages 0–0. SPE, 9 2003.

[2] Qiang Xiong and Arthur Jutan. Continuous optimization using a dynamic simplex method. *Chemical Engineering Science*, 58(16):3817–3828, 8 2003.

[3] Alejandro Marchetti, Grégory François, Timm Faulwasser, and Dominique Bonvin. Modifier Adaptation for Real-Time Optimization—Methods and Applications. *Processes*, 4(4):55, 12 2016.

[4] Weihua Gao and Sebastian Engell. Iterative set-point optimization of batch chromatography. *Computers & Chemical Engineering*, 29(6):1401–1409, 5 2005.

[5] Piotr Tatjewski. ITERATIVE OPTIMIZING SET-POINT CONTROL – THE BASIC PRINCIPLE REDESIGNED. *IFAC Proceedings Volumes*, 35(1):49–54, 2002.

[6] A. Marchetti, B. Chachuat, and D. Bonvin. Modifier-Adaptation Methodology for Real-Time Optimization. *Industrial & Engineering Chemistry Research*, 48(13):6022–6033, 7 2009.

[7] Weihua Gao, Simon Wenzel, and Sebastian Engell. A reliable modifier-adaptation strategy for real-time optimization. *Computers & Chemical Engineering*, 91:318–328, 8 2016.

[8] Sigurd Skogestad. Plantwide control: the search for the self-optimizing control structure. *Journal of Process Control*, 10(5):487–507, 10 2000.

[9] Ivar J. Halvorsen, Sigurd Skogestad, John C. Morud, and Vidar Alstad. Optimal Selection of Controlled Variables. *Industrial & Engineering Chemistry Research*, 42(14):3273–3284, 7 2003.

[10] Dinesh Krishnamoorthy, Bjarne Foss, and Sigurd Skogestad. Steady-state real-time optimization using transient measurements. *Computers & Chemical Engineering*, 115:34–45, 7 2018.

[11] José O.A. Matias and Galo A.C. Le Roux. Real-time Optimization with persistent parameter adaptation using online parameter estimation. *Journal of Process Control*, 68:195–204, 8 2018.

[12] G. François, B. Srinivasan, and D. Bonvin. Use of measurements for enforcing the necessary conditions of optimality in the presence of constraints and uncertainty. *Journal of Process Control*, 15(6):701–712, 9 2005.

[13] Stephen Boyd, Lin Xiao, Almir Mutapcic, and Jacob Mattingley. Notes on Decomposition methods, 2008.

[14] Hans P. Bieker, Olav Slupphaug, and Tor A. Johansen. Real-Time Production Optimization of Oil and Gas Production Systems: A Technology Survey. *SPE Production & Operations*, 22(04):382–391, 11 2007.

[15] Mark L. Darby, Michael Nikolaou, James Jones, and Doug Nicholson. RTO: An overview and assessment of current practice. *Journal of Process Control*, 21(6):874–884, 7 2011.

[16] Risvan Dirza and Sigurd Skogestad. Handling Interactive Systems in Primal-dual Feedback-optimizing Control. In *Foundations of Computer Aided Process Operations / Chemical Process Control*, pages 0–0, 2023.

[17] W Findeisen, F Bailey, M Brdys, K Malinowski, P Tatjewski, and A Wozniak. *Control and Coordination in Hierarchical Systems*. John Wiley and Sons, Chichester, UK, 1980.

[18] Ion Necoara, Bart De Schutter, Ton J. J. van den Boom, and Hans Hellendoorn. Robust control of constrained max-plus-linear systems. *International Journal of Robust and Nonlinear Control*, 19(2):218–242, 1 2009.

[19] Jinjun Zhuge and Marianthi G. Ierapetritou. Integration of Scheduling and Control with Closed Loop Implementation. *Industrial & Engineering Chemistry Research*, 51(25):8550–8565, 6 2012.

[20] Risvan Dirza, Alejandro Marquez-Ruiz, Leyla Özkan, and Carlos S. Mendez-Blanco. Integration of Max-Plus-Linear Scheduling and Control. In *Computer Aided Chemical Engineering*, volume 46, pages 1279–1284. 2019.

[21] Sigurd Skogestad. Control structure design for complete chemical plants. *Computers & Chemical Engineering*, 28(1-2):219–234, 1 2004.

[22] A Maarleveld and J.E. Rijnsdorp. Constraint Control on Distillation Columns. *Automatica*, 6:51–58, 1970.

[23] Johannes Jäschke and Sigurd Skogestad. Optimal controlled variables for polynomial systems. *Journal of Process Control*, 22(1):167–179, 1 2012.

[24] André D. Quelhas, Normando José Castro de Jesus, and José Carlos Pinto. Common vulnerabilities of RTO implementations in real chemical processes. *The Canadian Journal of Chemical Engineering*, 91(4):652–668, 4 2013.

[25] Maurício Câmara, André Quelhas, and José Pinto. Performance Evaluation of Real Industrial RTO Systems. *Processes*, 4(4):44, 11 2016.

[26] Rudolph Emil Kalman and Others. A new approach to linear filtering and prediction problems. *Journal of basic Engineering*, 82:35–45, 1960.

[27] A. Foss. Critique of chemical process control theory. *IEEE Transactions on Automatic Control*, 18(6):642–652, 12 1973.

[28] Manfred Morari, Yaman Arkun, and George Stephanopoulos. Studies in the synthesis of control structures for chemical processes: Part I: Formulation of the problem. Process decomposition and the classification of the control tasks. Analysis of the optimizing control structures. *AIChE Journal*, 26(2):220–232, 3 1980.

[29] Vidar Alstad and Sigurd Skogestad. Null Space Method for Selecting Optimal Measurement Combinations as Controlled Variables. *Industrial & Engineering Chemistry Research*, 46(3):846–853, 2007.

[30] Vinay Kariwala. Optimal Measurement Combination for Local Self-Optimizing Control. *Industrial & Engineering Chemistry Research*, 46(11):3629–3634, 5 2007.

[31] Ramprasad Yelchuru and Sigurd Skogestad. Convex formulations for optimal selection of controlled variables and measurements using Mixed Integer Quadratic Programming. *Journal of Process Control*, 22(6):995–1007, 7 2012.

[32] Vidar Alstad. *Studies on selection of controlled variables*. PhD thesis, Norwegian University of Science and Technology, 2005.

[33] Johannes Jäschke, Yi Cao, and Vinay Kariwala. Self-optimizing control – A survey. *Annual Reviews in Control*, 43:199–223, 2017.

[34] James J. Downs and Sigurd Skogestad. An industrial and academic perspective on plantwide control. *Annual Reviews in Control*, 35(1):99–110, 4 2011.

[35] Bala Srinivasan, Grégory François, and Dominique Bonvin. Comparison of Gradient Estimation Methods for Real-time Optimization. In E N Pistikopoulos, M C Georgiadis, and A C Kokossis, editors, *21st European Symposium on Computer Aided Process Engineering*, volume 29 of *Computer Aided Chemical Engineering*, pages 607–611. Elsevier, 2011.

[36] J. Abadie and J. Carpenter. *Generalization of the Wolfe reduced gradient method to the case of nonlinear constraints*. Academic Press, New York, 1969.

[37] Johannes Jäschke and Sigurd Skogestad. NCO tracking and self-optimizing control in the context of real-time optimization. *Journal of Process Control*, 21(10):1407–1416, 12 2011.

[38] Hirofumi Uzawa. Walras' tatonnement in the theory of exchange. *The Review of Economic Studies*, 27(3):182–194, 1960.

[39] Anders Rantzer. Dynamic dual decomposition for distributed control. In *2009 American Control Conference*, pages 884–888. IEEE, 2009.

[40] Dinesh Krishnamoorthy. A distributed feedback-based online process optimization framework for optimal resource sharing. *Journal of Process Control*, 97:72–83, 2021.

[41] Risvan Dirza, Sigurd Skogestad, and Dinesh Krishnamoorthy. Optimal Resource Allocation using Distributed Feedback-based Real-time Optimization. *IFAC-PapersOnLine*, 54(3):706–711, 2021.

[42] Dinesh Krishnamoorthy and Sigurd Skogestad. Real-Time Optimization as a Feedback Control Problem - A Review. *Computers & Chemical Engineering*, page 107723, 2 2022.

[43] Jorge Nocedal and Stephen J. Wright. *Numerical Optimization*. Springer, New York, NY, USA, second edition, 2006.

[44] Leon. S. Lasdon. *Optimization Theory for Large Systems*. Dover Publication Inc., New York, 1970.

[45] Philip Wolfe. A duality theorem for non-linear programming. *Quaterly of Applied Mathematics*, 19:239–244, 161.

[46] Sigurd Skogestad and Ian Postlethwaite. *Multivariable Feedback Control*. John Wiley and Sons, New York, NY, USA, 2005.

[47] Risvan Dirza, Dinesh Krishnamoorthy, and Sigurd Skogestad. Primal-dual Feedback-optimizing Control with Direct Constraint Control. In *14th International Symposium on Process System Engineering*, volume 49, pages 1153–1158. Elsevier, Kyoto, Japan, 6 2022.

[48] Risvan Dirza and Sigurd Skogestad. Online Feedback-based Optimization with Multi-input Direct Constraint Control. *IFAC-PapersOnline*, 55(7):149–154, 2022.

[49] B. Srinivasan, L.T. Biegler, and D. Bonvin. Tracking the necessary conditions of optimality with changing set of active constraints using a barrier-penalty function. *Computers & Chemical Engineering*, 32(3):572–579, 3 2008.

[50] B. Chachuat, A. Marchetti, and D. Bonvin. Process optimization via constraints adaptation. *Journal of Process Control*, 18(3-4):244–257, 3 2008.

[51] Dinesh Krishnamoorthy and Sigurd Skogestad. Systematic design of active constraint switching using selectors. *Computers & Chemical Engineering*, 143:107106, 12 2020.

[52] Sigurd Skogestad. Simple analytic rules for model reduction and PID controller tuning. *Journal of Process Control*, 13(4):291–309, 2003.

[53] Joel Andersson, Joris Gillis, Greg Horn, James B Rawlings, and Moritz Diehl. CasADi: a software framework for nonlinear optimization and optimal control. *Mathematical Programming Computation*, 11:1–36, 2019.

[54] Timur Bikmukhametov and Johannes Jäschke. First Principles and Machine Learning Virtual Flow Metering: A Literature Review. *Journal of Petroleum Science and Engineering*, 184:106487, 1 2020.

[55] Dinesh Krishnamoorthy, Esmaeil Jahanshahi, and Sigurd Skogestad. Feedback Real-Time Optimization Strategy Using a Novel Steady-state Gradient Estimate and Transient Measurements. *Industrial & Engineering Chemistry Research*, 58(1):207–216, 1 2019.

[56] Risvan Dirza and Sigurd Skogestad. Primal-Dual Feedback-optimizing Control with Override for Real-Time Optimization. *Journal of Process Control (under review)*, 2024.

[57] Sigurd Skogestad. Advanced control using decomposition and simple elements. *Annual Reviews in Control*, 56:100903, 2023.

[58] M.J. Fetkovich. The Isochronal Testing of Oil Wells. In *All Days*. SPE, 9 1973.

[59] Fathi Elldakli. Gas Lift System. *Petroleum & Petrochemical Engineering Journal*, 1(3), 2017.

[60] O.M. Aamo, G.O. Eikrem, H.B. Siahaan, and B.A. Foss. Observer design for multiphase flow in vertical pipes with gas-lift—theory and experiments. *Journal of Process Control*, 15(3):247–257, 4 2005.

[61] Jianbin Qiu, Wenqiang Ji, and Hak-Keung Lam. A New Design of Fuzzy Affine Model-Based Output Feedback Control for Discrete-Time Nonlinear Systems. *IEEE Transactions on Fuzzy Systems*, 31(5):1434–1444, 5 2023.

[62] M. Campos, H. Teixeira, F. Liporace, and M. Gomes. Challenges and problems with advanced control and optimization technologies. *IFAC Proceedings Volumes*, 42(11):1–8, 2009.

[63] B. J. T. Binder, D. K. M. Kufoalor, A. Pavlov, and T. A. Johansen. Embedded Model Predictive Control for an Electric Submersible Pump on a Programmable Logic Controller. In *2014 IEEE Conference on Control Applications (CCA)*, pages 579–585. IEEE, 10 2014.

[64] Stephen Boyd, Neal Parikh, Eric Chu, Borja Peleato, and Jonathan Eckstein. Distributed Optimization and Statistical Learning via the Alternating Direction Method of Multipliers. *Foundations and Trends® in Machine Learning*, 3(1):1–122, 2010.

[65] Risvan Dirza, Jose Matias, Sigurd Skogestad, and Dinesh Krishnamoorthy. Experimental validation of distributed feedback-based real-time optimization in a gas-lifted oil well rig. *Control Engineering Practice*, 126:105253, 9 2022.

[66] Simon Wenzel, Radoslav Paulen, Goran Stojanovski, Stefan Krämer, Benedikt Beisheim, and Sebastian Engell. Optimal resource allocation in industrial complexes by distributed optimization and dynamic pricing. *at - Automatisierungstechnik*, 64(6):428–442, 2016.

[67] Donald A. Walker. Walras's Theories of Tatonnement. *Journal of Political Economy*, 95:758–774, 1987.

[68] Sebastian Engell. Feedback control for optimal process operation. *Journal of Process Control*, 17(3):203–219, 3 2007.

[69] B. Chachuat, B. Srinivasan, and D. Bonvin. Adaptation strategies for real-time optimization. *Computers & Chemical Engineering*, 33(10):1557–1567, 10 2009.

[70] Bala Srinivasan and Dominique Bonvin. 110th Anniversary : A Feature-Based Analysis of Static Real-Time Optimization Schemes. *Industrial & Engineering Chemistry Research*, 58(31):14227–14238, 8 2019.

[71] Ioannis Tsamardinos, Laura E. Brown, and Constantin F. Aliferis. The max-min hill-climbing Bayesian network structure learning algorithm. *Machine Learning*, 65(1):31–78, 10 2006.

[72] Miroslav Krstić and Hsin-Hsiung Wang. Stability of extremum seeking feedback for general nonlinear dynamic systems. *Automatica*, 36(4):595–601, 4 2000.

[73] Kartik B. Ariyur and Miroslav Krstić. *Real-Time Optimization by Extremum-Seeking Control*. John Wiley & Sons, Inc., Hoboken, NJ, USA, 9 2003.

[74] G. François, B. Srinivasan, and D. Bonvin. Comparison of six implicit real-time optimization schemes. *Journal européen des systémes automatisés*, 46:291–305, 2012.

[75] R. E. Kalman and R. S. Bucy. New Results in Linear Filtering and Prediction Theory. *Journal of Basic Engineering*, 83(1):95–108, 3 1961.

[76] Rinaldo A. Jose and Lyle H. Ungar. Pricing interprocess streams using slack auctions. *AIChE Journal*, 46(3):575–587, 3 2000.

[77] Rubén Martí, Daniel Navia, Daniel Sarabia, and César De Prada. Shared Resources Management by Price Coordination. In *Computer Aided Chemical Engineering*, pages 902–906. 2012.

[78] Goran Stojanovski, Lukas Maxeiner, Stefan Kramer, and Sebastian Engell. Real-time shared resource allocation by price coordination in an integrated petrochemical site. In *2015 European Control Conference (ECC)*, pages 1498–1503. IEEE, 7 2015.

[79] Shaghayegh Nazari, Christian Sonntag, Goran Stojanovski, and Sebastian Engell. A Modelling, Simulation, and Validation Framework for the Distributed Management of Large-scale Processing Systems. In *Computer Aided Chemical Engineering*, pages 269–274. 2015.

[80] Vidar Gunnerud and Bjarne Foss. Oil production optimization—A piecewise linear model, solved with two decomposition strategies. *Computers & Chemical Engineering*, 34(11):1803–1812, 11 2010.

[81] Dinesh Krishnamoorthy, Marco Aurelio Aguiar, Bjarne Foss, and Sigurd Skogestad. A Distributed Optimization Strategy for Large Scale Oil and Gas Production Systems. In *2018 IEEE Conference on Control Technology and Applications (CCTA)*, pages 521–526. IEEE, 8 2018.

[82] Dinesh Krishnamoorthy, Carlo Valli, and Sigurd Skogestad. Real-time Optimal Resource Allocation in an Industrial Symbiotic Network using Transient Measurements [*]. In *2020 American Control Conference (ACC)*, pages 3541–3546. IEEE, 7 2020.

[83] J. Matias, J. P. C. Oliveira, G. A. C. Le Roux, and J. Jaschke. Steady-state Real-time Optimization Using Transient Measurements on an Experimental Rig. *Journal of Process Control*, 115:181–196, 9 2022.

[84] Risvan Dirza, Sigurd Skogestad, and Dinesh Krishnamoorthy. Real-Time Optimal Resource Allocation and Constraint Negotiation Applied to A Subsea Oil Production Network. In *SPE Annual Technical Conference and Exhibition*, pages 21–23. SPE, 9 2021.

[85] Michael Baldea and Prodromos Daoutidis. Control of integrated process networks—A multi-time scale perspective. *Computers & Chemical Engineering*, 31(5):426–444, 2007.

[86] Eric Walter and Luc Pronzato. *Identification of parametric models: from experimental data*. Springer Verlag, 1997.

[87] Rick Bitter, Taqi Mohiuddin, and Matt Nawrocki. *LabVIEW: Advanced programming techniques*. Crc Press, 2006.

[88] Risvan Dirza and Sigurd Skogestad. A Systematic Procedure of Implementing Primal-Dual with Direct Constraint Control. *(In preparation)*, 2022.

[89] Risvan Dirza and Sigurd Skogestad. Systematic Pairing Selection for Economic-oriented Constraint Control. In *32nd European Symposium on Computer Aided Process Engineering*, volume 51, pages 1249–1254. Elsevier, Touluse, France, 6 2022.

[90] Yildiray Yildiz, Ilya V. Kolmanovsky, and Diana Acosta. A control allocation system for automatic detection and compensation of phase shift due to actuator rate limiting. In *Proceedings of the 2011 American Control Conference*, pages 444–449. IEEE, 6 2011.

[91] Risvan Dirza, Md Rizwan, Sigurd Skogestad, and Dinesh Krishnamoorthy. Real-time Optimal Resource Allocation using Online Primal Decomposition. *IFAC-PapersOnLine*, 55(21):31–36, 2022.

[92] Lærke Skov Hansen, Simon Pedersen, and Petar Durdevic. Multi-Phase Flow Metering in Offshore Oil and Gas Transportation Pipelines: Trends and Perspectives. *Sensors*, 19(9), 2019.

[93] Yi-Ming Wei, Qing-huai Hu, and Ying Fan. Mathematical model for the optimization of the allocation of nonferrous raw materials in China. *Computers & Industrial Engineering*, 46(2):293–303, 4 2004.

[94] Vegard Aas, Risvan Dirza, Dinesh Krishnamoorthy, and Sigurd Skogestad. A comparative study of distributed feedback-optimizing control strategies. In *33rd European Symposium on Computer Aided Process Engineering*, volume 52, pages 613–618. 2023.

[95] Risvan Dirza, Vegard Aas, Sigurd Skogestad, and Dinesh Krishnamoorthy. A Comparative Study of Distributed Feedback-optimizing Control Schemes: An Experimental Validation. *IEEE Transactions on Control Systems Technology (under review)*, 2024.

[96] Dinesh Krishnamoorthy and Sigurd Skogestad. Linear Combination of Gradients as Optimal Controlled Variables. In *Computer aided chemical engineering*, volume 48, pages 1237–1242. 2020.

[97] K.J. Arrow, L. Hurwicz, and H. Uzawa. *Studies in Linear and Non-Linear Programming*. Stanford University Press, Redwood City, 1958.

[98] Goran Banjac, Felix Rey, Paul Goulart, and John Lygeros. Decentralized Resource Allocation via Dual Consensus ADMM. In *2019 American Control Conference (ACC)*, pages 2789–2794. IEEE, 7 2019.

[99] J. M. Maestre, D. Munoz de la Pena, and E. F. Camacho. Distributed MPC based on a cooperative game. In *Proceedings of the 48h IEEE Conference on Decision and Control (CDC) held jointly with 2009 28th Chinese Control Conference*, pages 5390–5395. IEEE, 12 2009.

[100] Md Arifin Arif, Mandoye Ndoye, Gregory V. Murphy, and Kennedy Aganah. A cooperative game theory algorithm for distributed reactive power reserve optimization and voltage profile improvement. In *2017 North American Power Symposium (NAPS)*, pages 1–6. IEEE, 9 2017.

[101] R. Halin. A theorem on n-connected graphs. *Journal of Combinatorial Theory*, 7(2):150–154, 9 1969.

[102] Alan C. Hindmarsh, Peter N. Brown, Keith E. Grant, Steven L. Lee, Radu Serban, Dan E. Shumaker, and Carol S. Woodward. SUNDIALS. *ACM Transactions on Mathematical Software*, 31(3):363–396, 9 2005.

[103] S. Wenzel, F. Riedl, and S. Engell. An efficient hierarchical market-like coordination algorithm for coupled production systems based on quadratic approximation. *Computers & Chemical Engineering*, 134:106704, 3 2020.

[104] Towfiq Rahman, Zhihua Qu, and Toru Namerikawa. Improving Rate of Convergence via Gain Adaptation in Multi-Agent Distributed ADMM Framework. *IEEE Access*, 8:80480–80489, 2020.

[105] Risvan Dirza, Edmary Altamiranda, and Sigurd Skogestad. Self-Optimizing Control for Recirculated Gas lifted Subsea Oil Well Production. *IFAC Proceedings Volumes (to appear)*, 2024.

[106] Dinesh Krishnamoorthy, Bjarne Foss, and Sigurd Skogestad. Real-Time Optimization under Uncertainty Applied to a Gas Lifted Well Network. *Processes*, 4(4):52, 12 2016.

[107] J.T. Gravdahl and Egeland O. *Compressor Surge and Rotating Stall: Modeling and Control*. Springer, 1998.

[108] A. Cortinovis, H.J. Ferreau, D. Lewandowski, and M. Mercangöz. Experimental evaluation of MPC-based anti-surge and process control for electric driven centrifugal gas compressors. *Journal of Process Control*, 34:13–25, 10 2015.

[109] Christoph J. Backi, Brian A. Grimes, and Sigurd Skogestad. A Control- and Estimation-Oriented Gravity Separator Model for Oil and Gas Applications Based upon First-Principles. *Industrial & Engineering Chemistry Research*, 57(21):7201–7217, 5 2018.

[110] Predrag Milosavljevic, Alejandro G. Marchetti, Andrea Cortinovis, Timm Faulwasser, Mehmet Mercangöz, and Dominique Bonvin. Real-time optimization of load sharing for gas compressors in the presence of uncertainty. *Applied Energy*, 272:114883, 8 2020.

[111] P Mehdizadeh. Worldwide Multiphase and Wet Gas Metering Installations. Technical report, Production Technology Report, 3 2007.

[112] Vinay Kariwala and Yi Cao. Bidirectional branch and bound for controlled variable selection. Part II: Exact local method for self-optimizing control. *Computers & Chemical Engineering*, 33(8):1402–1412, 8 2009.

[113] Byron Tasseff, Carleton Coffrin, Andreas Wachter, and Carl D. Laird. Exploring Benefits of Linear Solver Parallelism on Modern Nonlinear Optimization Applications. *arXiv: Optimization and Control*, 2019.

[114] Maximilian Cegla, Aleksandra Fage, Simon Kemmerling, and Sebastian Engell. Experimental Application of Real-Time Optimization with Modifier Adaptation and Quadratic Approximation to a Reactive Extrusion Process. *IFAC-PapersOnLine*, 56(2):6150–6155, 2023.

[115] E. M. Greitzer. Surge and Rotating Stall in Axial Flow Compressors—Part I: Theoretical Compression System Model. *Journal of Engineering for Power*, 98(2):190–198, 4 1976.

Appendices

Appendix A

Interactive Gas Lift 1

A.1 Simple gas-lift system model

The mass balance equations are as follows:

$$\dot{m}_{gl} = w_{co,gl} - \sum_{i=1}^{N} w_{gl,i} \tag{A.1a}$$

$$\dot{m}_{ga,i} = w_{gl,i} - w_{iv,i} \tag{A.1b}$$

$$\tag{A.1c}$$

where m_{gl} is the mass of gas inside the supply line from gas lift compressor discharge point up to the gas-lift choke valve for all well, $m_{ga,i}$ is the mass of gas of each well inside the annulus from gas-lift choke valve up to the injection valve for each well, $w_{co,gl}$ is gas mass flow rate supplied by a gas-lift compressor, $w_{gl,i}$ is gas mass flow rate of each well injected to the annulus, and $w_{iv,i}$ is gas mass flow rate of each well injected to well tubing.

The mass flow rate equations are as follows:

$$w_{gl,i} = u_{gl,i}.c_{gl,i}\sqrt{\rho_{c,gl}\left(p_{co,gl} - p_{ai,i}\right)} \tag{A.2a}$$

$$w_{iv,i} = u_{iv,i}.c_{iv,i}\sqrt{\rho_{ai,i}\left(p_{ai,i} - p_{wi,i}\right)} \tag{A.2b}$$

where $c_{gl,i}$ is the valve coefficient of gas-lift choke valve, $c_{iv,i}$ is the valve coefficient of injection valve, $\rho_{c,gl}$ is gas mass density in gas-lift supply line, $\rho_{ai,i}$ is gas mass density in annulus of each well, $p_{co,gl}$ is gas pressure in supply line, $p_{ai,i}$ is gas pressure inside annulus, $p_{wi,i}$ is injection pressure inside well tubing, $u_{gl,i}$ is the valve opening of gas-lift choke valve and $u_{iv,i}$ is the valve opening of injection valve. If required, $w_{gl,i}$ and/or $w_{iv,i}$ can be controlled by $u_{gl,i}$ and/or $u_{iv,i}$ as manipulated variables.

Given $w_{c,gl}$, supplied by a gas-lift compressor, $u_{gl,i}$ gives splitting ratio to each well. Traditionally, $u_{gl,i}$ is set manually based on well testing data that estimate Gas-to-oil ratio. Meanwhile, $u_{iv,i}$ is set constant as it typically represents pressure drop from annulus to injection point in the tubing.

The gas density are defined as follows:

$$\rho_{c,gl} = \frac{M_w}{R.T_{c,gl}} p_{co,gl} \tag{A.3a}$$

$$\rho_{ai,i} = \frac{M_w}{R.T_{ai,i}} p_{ai,i} \tag{A.3b}$$

where M_w is molecular weight, R is Reynold number, $T_{c,gl}$ and $T_{ai,i}$ are the temperature of the gas inside gas supply line and annulus in each well respectively.

The pressure are defined as follows:

$$p_{co,gl} = \frac{R.T_{c,gl}}{M_w} \frac{m_{gl}}{\pi.r_{gl}^2.L_{gl}} \tag{A.4a}$$

$$p_{ai,i} = \left(\frac{R.T_{ai,i}}{M_w} + g.H_{ai,i} \right) \frac{m_{ga}}{\pi.(r_{ai,i}^2 - r_{wi,i}^2).L_{ai,i}} \tag{A.4b}$$

where r_{gl}, $r_{ai,i}$ and $r_{wi,i}$ are the radius of gas-lift supply line, annulus and well tubing respectively. L_{gl}, and $L_{ai,i}$ are the length of gas-lift supply line, and annulus respectively. g is the gravity force, and $H_{ai,i}$ is the height of annulus.

A.2 Simple oil production well model

The mass balance equations are as follows:

$$\dot{m}_{gt,i} = w_{rg,i} + w_{iv,i} - w_{pg,i} \tag{A.5a}$$

$$\dot{m}_{ot,i} = w_{ro,i} - w_{po,i} \tag{A.5b}$$

where $m_{gt,i}$ is the mass of gas inside well tubing, $m_{ot,i}$ is the mass of oil inside well tubing, $w_{rg,i}$ is gas mass flow rate from reservoir, $w_{ro,i}$ is oil mass flow rate from reservoir, $w_{pg,i}$ is produced gas mass flow rate, and $w_{po,i}$ is produced oil mass flow rate.

The mass flow rate equations are as follows:

$$w_{pc,i} = u_{pc,i}.c_{pc,i} \sqrt{\rho_{mix,i} \left(p_{wh,i} - p_m \right)} \tag{A.6a}$$

$$w_{rg,i} = GOR_i.w_{ro,i} \tag{A.6b}$$

$$w_{pg,i} = \frac{m_{gt,i}}{m_{gt,i} + m_{ot,i}}.w_{pc,i} \tag{A.6c}$$

$$w_{ro,i} = PI_i.\left(p_{res,i} - p_{bh,i} \right) \tag{A.6d}$$

$$w_{po,i} = \frac{m_{ot,i}}{m_{gt,i} + m_{ot,i}}.w_{pc,i} \tag{A.6e}$$

where $w_{pc,i}$ is produced hydrocarbon (both oil and gas) mass flow rate, $c_{pc,i}$ is the valve coefficient of production choke valve, $\rho_{mix,i}$ is mixed mass density in the well tubing, $p_{res,i}$ is reservoir pressure of each well, $p_{bh,i}$ is bottomhole pressure of each well, $p_{wh,i}$ is wellhead pressure of each well, p_m is manifold pressure, GOR_i is gas-to-oil ratio of each well, and PI_i is well productivity index.

Usually, $u_{pc,i}$ is fully open to maximize the production flow. However, this valve can be used to control or even close the well in case of significant reservoir disturbance i.e., gas coning.

The mass density is defined as follows:

$$\rho_{mix,i} = \frac{m_{gt,i} + m_{ot,i} - \rho_{o,i}.\pi.r_{bh,i}^2 L_{bh,i}}{\pi.r_{wi,i}^2 L_{wi,i}} \tag{A.7a}$$

where $\rho_{o,i}$ is oil density, $r_{bh,i}$ is the radius of bottomhole, and $L_{bh,i}$ and $L_{wi,i}$ are the length of bottomhole, and well tubing respectively.

The pressure are defined as follows:

$$p_{wh,i} = \frac{R.T_{wi,i}}{M_w} \frac{m_{gt,i}}{\pi.r_{wi,i}^2.L_{wi,i} + \pi.r_{bh,i}^2.L_{bh,i} - \frac{m_{ot,i}}{\rho_{o,i}}} - \Delta p_{fric,wi,i} \tag{A.8a}$$

$$p_{wi,i} = p_{wh,i} + \frac{m_{gt,i} + m_{ot,i} - \rho_o.\pi.r_{bh,i}^2.L_{bh,i}}{\pi.r_{wi,i}^2.L_{wi,i}}.g.H_{wi,i} \tag{A.8b}$$

$$p_{bh,i} = p_{wi,i} + \rho_{o,i}.g.H_{bh,i} \tag{A.8c}$$

$$p_m = p_{rh} + g.H_r.\frac{m_{or} + m_{gr}}{\pi.r_r^2.L_r} \tag{A.8d}$$

where $T_{wi,i}$ is the temperature in well tubing, $\Delta p_{fric,wi,i}$ is pressure drop due to friction, $H_{wi,i}$ is the height of well tubing from well head to injection point, and $H_{bh,i}$ is the height from injection point to the bottomhole.

The manifold pressure, p_m depends on riser head pressure, p_{rh}, height of riser, H_r, mass of oil, m_{or} and gas m_{gr} in the riser, and radius, r_r and length L_r of riser, given by simple riser system model.

The assumptions in the aforementioned equations are:

1. The oil density $\rho_{o,i}$ is relatively constant.
2. The mass of oil computed for mixture density excludes mass of oil in the bottomhole.

A.3 Simple riser model

The mass balance equations are as follows:

$$\dot{m}_{gr} = \sum_{i=1}^{N} w_{pg,i} - w_{gr} \tag{A.9a}$$

$$\dot{m}_{or} = \sum_{i=1}^{N} w_{po,i} - w_{or} \tag{A.9b}$$

where m_{gr} is the mass of gas inside the riser, m_{or} is the mass of oil inside the riser, w_{gr} is gas mass flow rate in the riser, and w_{or} is oil mass flow rate in the riser.

The mass flow rate equations are as follows:

$$w_{pr,i} = u_{pr}.c_{pr}\sqrt{\rho_r\left(p_{rh} - p_{sep}\right)} \tag{A.10a}$$

$$w_{gr,i} = \frac{m_{gr}}{m_{gr} + m_{or}}.w_{pr} \tag{A.10b}$$

$$w_{or,i} = \frac{m_{or}}{m_{gr} + m_{or}}.w_{pr} \tag{A.10c}$$

where w_{pr} is total produced hydrocarbon (both oil and gas) mass flow rate in the riser, c_{pr} is the valve coefficient of riser choke valve, ρ_r is mass density in the riser, p_{rh} is riserhead pressure, and p_{sep} is separator pressure.

The separator pressure, p_{sep} depends on gas volume capacity in the separator. Usually, u_{pr} is set constant as it typically represents pressure drop from manifold to the riser head.

The mass density is defined as follows:

$$\rho_r = \frac{m_{gr} + m_{or}}{\pi . r_r^2 L_r} \tag{A.11a}$$

where r_r is the radius of the riser and L_r is the length of the riser.

The pressure are defined as follows:

$$p_{rh} = \frac{R.T_r}{M_w} \frac{m_{gr}}{\pi . r_r^2 . L_r} - \Delta p_{fric,r} \tag{A.12a}$$

where T_r is the temperature in the riser, $\Delta p_{fric,r}$ is pressure drop due to friction in the riser, and L_r is the length of riser from riserhead to manifold.

A.4 Simple separator model

The mass balance equations are as follows:

$$\dot{m}_{gs} = w_{gr} - w_{gs} \tag{A.13a}$$

$$\dot{m}_{os} = w_{or} - w_{os} \tag{A.13b}$$

where m_{gs} is the mass of gas inside the separator, m_{os} is the mass of oil inside the separator, w_{gs} is gas mass out flow rate from separator, and w_{os} is oil mass out flow rate from the separator.

If some of the gas are used for gas-lift injection, then

$$w_{gs} = w_{ci,gl} + w_{c,ge} \tag{A.14a}$$

where $w_{c,ge}$ is gas mass flow rate exported to gas processing facilities, and $w_{ci,gl}$ is gas mass flow rate exported to gas lift compressor,

The gas flow rate $w_{ci,gl}$ and $w_{c,ge}$ are influenced by compressor capacity. Meanwhile, the oil flow rate w_{os} is influenced by pump capacity.

The pressure is defined as dynamic model as follows:

$$\dot{p}_{sep} = \frac{R.T_{sep}}{V_{gs} M_w} (\dot{m}_{gs}) + \frac{p_{sep}}{V_{gs} \rho_o} (\dot{m}_{os}) \tag{A.15a}$$

$$p_{os} = p_{sep} + \rho_o . g . h_{os} \tag{A.15b}$$

where T_{sep} is the temperature in the separator, p_{os} is pressure in the oil outlet of the separator, and h_{os} is the height of oil level in the separator. Note that the separator pressure p_{sep} is influenced by both the gas in-and out mass flow rate, and the gas volume inside the separator that is influenced by both the oil in-and out mass flow rate.

The height of oil level in the separator is defined as follows:

$$\dot{h}_{os} = \frac{1}{\rho_o}\left(\dot{m}_{os}\right)\frac{1}{2.L_{sep}.\sqrt{h_{os}\left(2.r_{sep}-h_{os}\right)}} \tag{A.16a}$$

(A.16b)

where L_{sep} is the length of separator and r_{sep} is the radius of separator. Here, we consider horizontal separator.

A.5 Simple compressor model

For the sake of simplification, we consider an isothermal compressor as follows:

$$w_{co,gl} = Pow_{c,gl}.\frac{M_w}{R.T_{c,gl}.\Phi_{c,gl}} \tag{A.17a}$$

$$w_{co,ge} = Pow_{c,ge}.\frac{M_w}{R.T_{c,ge}.\Phi_{c,ge}} \tag{A.17b}$$

where $Pow_{c,gl}$ is the power required to deliver gas flow rate $w_{co,gl}$, with pressure ratio $\Phi_{c,gl}$ at temperature $T_{c,gl}$ And $Pow_{c,ge}$ is the power required to deliver gas flow rate $w_{co,ge}$, with pressure ratio $\Phi_{c,ge}$ at temperature $T_{c,ge}$. Note that, we do not consider recycle line in this model. Thus, $w_{co,gl} = w_{ci,gl}$, and $w_{co,ge} = w_{ci,ge}$.

The pressure ratios are defined as follows:

$$\Phi_{c,gl} = \ln\frac{p_{co,gl}}{p_{ci,gl}} \tag{A.18a}$$

$$\Phi_{c,ge} = \ln\frac{p_{co,ge}}{p_{sep}} \tag{A.18b}$$

where $p_{ci,gl} = p_{sep}$ if some of the gas separated in the separator are used for gas-lift injection. If we consider fixed compressor, then Eq. (A.18) are constant.

A more comprehensive compressor model is also considered when it is necessary. In this case, we consider the well-known compressor model originally developed by [115] and later extended by [107]. This model was also used in control and optimization investigation by [108, 110].

The pressure are defined as dynamic compressor model as follows:

$$\dot{p}_{s,gl} = C_{1,gl}\left(w_{ci,gl} + w_{rec,gl} - w_{c,gl}\right) \tag{A.19a}$$

$$\dot{p}_{s,ge} = C_{1,ge}\left(w_{ci,ge} + w_{rec,ge} - w_{c,ge}\right) \tag{A.19b}$$

$$\dot{p}_{d,gl} = C_{2,gl}\left(w_{c,gl} - w_{rec,gl} - w_{co,gl}\right) \tag{A.19c}$$

$$\dot{p}_{d,ge} = C_{2,ge}\left(w_{c,ge} - w_{rec,ge} - w_{co,ge}\right) \tag{A.19d}$$

where $p_{s,gl}$, $p_{s,ge}$, $p_{d,gl}$, and $p_{d,ge}$ are suction pressure of gas lift compressor, suction pressure of gas export compressor, downstream pressure of gas lift compressor, and downstream pressure of gas export compressor, respectively. $w_{ci,gl}$, $w_{ci,ge}$, $w_{co,gl}$, $w_{co,ge}$, $w_{c,gl}$, $w_{c,ge}$, $w_{rec,gl}$, and $w_{rec,ge}$ are the mass flow rate of incoming gas in gas-lift compressor, incoming gas in gas-export compressor, outgoing gas in gas-lift compressor, outgoing gas

in gas-export compressor, gas inside gas-lift compressor, gas inside gas-export compressor, recycle gas inside gas-lift compressor, and recycle gas inside gas-export compressor, respectively. $C_{1,gl}$, $C_{1,ge}$, $C_{2,gl}$, and $C_{2,ge}$ are model coefficients.

The gas mass flow rate inside the compressor is also defined as dynamic compressor model as follows:

$$\dot{w}_{c,gl} = C_{3,gl}\left(p_{s,gl}\Pi_{gl} - p_{d,gl}\right) \tag{A.20a}$$

$$\dot{w}_{c,ge} = C_{3,ge}\left(p_{s,ge}\Pi_{ge} - p_{d,ge}\right) \tag{A.20b}$$

where Π_{gl} and Π_{ge} are the pressure ratio of gas-lift compressor and gas-export compressor respectively. $C_{3,gl}$, and $C_{3,ge}$ are model coefficients.

Furthermore, we have the following relationships:

$$w_{ci,gl} = u_{ci,gl}c_{ci,gl}\sqrt{\rho_{gl}\left(p_{ci,gl} - p_{s,gl}\right)} \tag{A.21a}$$

$$w_{ci,ge} = u_{ci,ge}c_{ci,ge}\sqrt{\rho_{ge}\left(p_{ci,ge} - p_{s,ge}\right)} \tag{A.21b}$$

Appendix B

Interactive Gas Lift 2

For this work, the notation shown in Table B.1-B.2 are used. This is the same notation as used in the Matlab file, but it is slightly different from the somewhat simplified notation that we have used in Chapter 3, for example, in Chapter 3, we write w_l instead of w_{gl} for the gas lift mass flow.

We assume constant oil density, and hence can express the rate of change in each well tubing, and in the riser, which are described by the following differential equations.

$$\dot{V}_{ot,i} = q_{or,i} - q_{ot,i} \tag{B.1a}$$

$$\dot{V}_{oe} = \sum_{i=1}^{4} q_{ot,i} - q_{oe} \tag{B.1b}$$

We assume ideal gas behavior for constant temperature, the pressure dynamics in each annulus, well tubing, and in the riser can be written as

$$\dot{p}_{a,i} = \frac{\frac{\rho_{ga,i} R T_{a,i}}{M_g} \left(q_{gl,i} - q_{ga,i}\right)}{V_a} \tag{B.2a}$$

$$\dot{p}_{h,i} = \frac{\frac{\rho_{gt,i} R T_{t,i}}{M_g} \left(q_{ga,i} + q_{gr,i} - q_{gt,i}\right) + p_{h,i} \dot{V}_{ot,i}}{V_{gt,i}} \tag{B.2b}$$

$$\dot{p}_e = \frac{\frac{\rho_{ge} R T_e}{M_g} \left(\sum_{i=1}^{4} q_{gt,i} - q_{ge}\right) + p_e \dot{V}_{oe}}{V_{ge}} \tag{B.2c}$$

The volume of gas inside annulus, well and riser tubing are given by,

$$V_{ga,i} = \pi r_{a,i}^2 L_{a,i} - \pi r_{t,i}^2 L_{t,i} \tag{B.3a}$$

$$V_{gt,i} = V_{t,i} - V_{ot,i} \tag{B.3b}$$

$$V_{ge,i} = V_{e,i} - V_{oe,i} \tag{B.3c}$$

The fluid velocities are given by,

$$v_{ft,i} = \frac{q_{fh,i}}{A_{t,i}} \tag{B.4a}$$

$$v_{fe} = \frac{q_{fe}}{A_e} \tag{B.4b}$$

The additional pressures are calculated by,

$$p_{b,i} = p_{h,i} + \Delta p_{xt,i} + \Delta p_{yt,i} \tag{B.5a}$$

$$\Delta p_{xt,i} = k \rho_{ft,i} L_{t,i} \tag{B.5b}$$

$$\Delta p_{yt,i} = \rho_{ft,i} w_{t,i} \left(\frac{L_{t,i} v_{ft,i}^2}{2 A_{t,i}} \right) \tag{B.5c}$$

$$p_{j,i} = p_{b,i} - k \rho_{ft,i} L_{j,i} - \rho_{ft,i} w_{t,i} \left(\frac{L_{j,i} v_{ft,i}^2}{2 A_{t,i}} \right) \tag{B.5d}$$

$$p_m = p_e + \Delta p_{xe} + \Delta p_{ye} \tag{B.5e}$$

$$\Delta p_{xe} = k \rho_{fe} L_e \tag{B.5f}$$

$$\Delta p_{ye} = \rho_{fe} w_e \left(\frac{L_e v_{fe}^2}{2 A_e} \right) \tag{B.5g}$$

The volumetric flow rates are given by,

$$q_{gl,i} = u_{gl,i} c_{gl,i} \sqrt{\max\left(0, \left(p_s^2 - p_{a,i}^2\right)\right)} \tag{B.6a}$$

$$q_{ga,i} = c_{ga,i} \sqrt{\max\left(0, \left(p_{a,i}^2 - p_{j,i}^2\right)\right)} \tag{B.6b}$$

$$q_{or,i} = \left(1 - \left(\frac{p_{b,i}}{p_{r,i}}\right)^2\right)^{n_i} Q_i \tag{B.6c}$$

$$q_{gr,i} = GOR_i . q_{or,i} \tag{B.6d}$$

$$q_{fh,i} = c_{fh,i} \sqrt{\max\left(0, \left(p_{h,i}^2 - p_m^2\right)\right)} \tag{B.6e}$$

$$q_{ot,i} = \frac{V_{ot,i}}{V_{t,i}} . q_{fh,i} \tag{B.6f}$$

$$q_{gt,i} = \frac{V_{gt,i}}{V_{t,i}} . q_{fh,i} \tag{B.6g}$$

$$q_{fe} = c_{fe} \sqrt{\max\left(0, \left(p_e^2 - p_d^2\right)\right)} \tag{B.6h}$$

$$q_{oe} = \frac{V_{oe}}{V_e} . q_{fe} \tag{B.6i}$$

$$q_{ge} = \frac{V_{ge}}{V_e} . q_{fe} \tag{B.6j}$$

The mixture densities are given by,

$$\rho_{ft,i} = \frac{V_{gt,i}}{V_{t,i}}\rho_{gt,i} + \frac{V_{ot,i}}{V_{t,i}}\rho_{ot,i} \tag{B.7a}$$

$$\rho_{fe} = \frac{V_{ge}}{V_e}\rho_{ge} + \frac{V_{oe}}{V_e}\rho_{oe} \tag{B.7b}$$

As seen from (B.1a) - (B.7b), the gas lifted well is modelled as a semi-explicit index-1 DAE system of the form with the set of differential equations (B.1a) - (B.2c) and the set of algebraic equations (B.3a) - (B.7b).

Table B.1: Notation used in Appendix and Matlab model-1

Notation	Description
i	well index
u	choke valve opening
f	fluid (both oil and gas)
o	oil
g	gas
x	hydro static
y	friction
ga	gas - annulus
ge	gas - riser (export)
gl	gas - lift supply
gr	gas - reservoir
gt	gas - well tubing
oe	oil - riser (export)
or	oil - reservoir
ot	oil - well tubing
a	annulus
b	bottomhole
d	separator
e	riser (-head)
h	wellhead
j	gas lift injection point
m	manifold
r	reservoir
s	gas lift supply line
t	well tubing

Table B.2: Notation used in Appendix and Matlab model-2

Notation	Description
A	cross section area
GOR	gas oil ratio
L	length
M	molar mass
Q	maximum reservoir oil volumetric rate
R	gas constant
T	temperature
V	volume
c	valve coefficient
k	gravity
n	flow exponent coefficient
p	pressure
q	volumetric flow rate
v	velocity
w	friction factor
ρ	density