On enforcing the necessary conditions of optimality under plant-model mismatch - what to measure and what to adapt?

THÈSE N^{O} 8803 (2018)

PRÉSENTÉE LE 19 OCTOBRE 2018

À LA FACULTÉ DES SCIENCES ET TECHNIQUES DE L'INGÉNIEUR

LABORATOIRE D'AUTOMATIQUE - COMMUN

PROGRAMME DOCTORAL EN CHIMIE ET GÉNIE CHIMIQUE

ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

POUR L'OBTENTION DU GRADE DE DOCTEUR ÈS SCIENCES

PAR

Martand SINGHAL

acceptée sur proposition du jury:

Prof. V. Hatzimanikatis, président du jury
Prof. D. Bonvin, Dr T. Faulwasser, directeurs de thèse
Prof. M. Fikar, rapporteur
Dr G. Francois, rapporteur
Prof. M. Bierlaire, rapporteur

ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE

Suisse
2018

When an idea exclusively occupies the mind, it is transformed into an actual physical or mental state.
— Swami Vivekananda

To the courageous

Acknowledgements

This thesis is an outcome of the five years of research journey that I began here in the Automatic Control Lab (LA). This thesis has been shaped by my interactions with numerous people that I met before and during this journey.

At first, I thank my thesis director Professor Dominique Bonvin for his guidance and patience throughout this time. He has moulded my approach to research and taught me how to observe things from the bird's view. I am always inspired by his leadership and organizational skills. I thank my thesis co-director Dr. Timm Faulwasser because of whom my technical writing and presentation skills improved tremendously. I also thank Dr. Alejandro Marchetti for his insightful discussions on RTO. His curious nature has pushed me to think in different dimensions that improved my overall understanding of the subject matter.

LA is kept functional by its staff members that form its backbone. My thanks to Christophe, Francis and Norbert for ensuring that I get everything that is required to run the projects. It is impossible to imagine LA without Ruth and Sandra that kept the lab atmosphere lively. Thanks to Ruth, Francine, Eva, Margot and Nicole for their administrative support. My thanks to all the professors in the lab including Professor Longchamp, Professor Colin Jones, and Professor Giancarlo Trecate Ferrari for the wonderful lab atmosphere and working conditions. I thank Dr. Alireza Karimi for helping me out during the teaching duties and for explaining about the estimation and control theory whenever I asked. My special thanks to Philippe for his engaging discussions that always turn to abstraction at some point or the other.

I thank Professor Vassily Hatzimanikatis for all the help during the last period of my doctoral studies. He ensured that I could finish my thesis smoothly by providing the administrative and financial support. I also thank Dr. Ljubisa Miskovic for his guidance and support towards the end of my doctoral studies.

I thank all my collaborators with whom I did projects including all the master students and colleagues. My thanks to the Dortmund group for the nice collaboration, I really enjoyed my time working with them. Special thanks to Afaq and Matthieu for their dedication and for being excellent team players.

Acknowledgements

PhD, like any other aspect of life, has its ups and downs. I thank Milan, Zlatko, Ehsan, Christoph, Predrag, Georgios, Jean-Hubert and Philippe for lending me an ear whenever I needed. Thanks to Predrag, Tafarel, Diogo, Gene and Sean for the long research discussions that we had in all these years. Thanks to Christoph, Ehsan, Tafarel, Peter, Jean-Hubert, Sanket, Milan, Georgios, Predrag, Harsh, Altug, Diogo, Sriniketh, Zlatko, Mahdieh, Pulkit, Luca, Francisco, Ivan, Faran, Sohail, Mustafa, Tomasz, Michele and Andrea for the fun office atmosphere and all the enjoyable moments we spent in the office, during conferences and outings.

My doctoral studies in Lausanne has been such a great experience inside and outside the PhD life. I have met numerous wonderful people with whom I have exchanged ideas, had enriching cross-cultural experiences, carved new friendships, shared laughs and did various activities. Special thanks to Ioannis, Milan, Giulia, Evelyne, Carla, Maria, Daria, Aamani and Niko for their love and friendship. I am also grateful to all my friends back in India and abroad who have always been there for me.

Finally, I thank my family for everything. My mother Archana and my sister Richa, my deepest gratitude for their unconditional love and support. I thank Cécile, without her the life would not have been the same. She has been a constant source of love, encouragement and support. I also thank Denise and Jean-Marc for their love and affection. I found a family in them and they make Lausanne feel like home.

Lausanne, August 2018 Martand Singhal

Abstract

Industrial processes are run with the aim of maximizing economic profit while simultaneously meeting process-critical constraints. To this end, model-based optimization can be performed to ensure optimal plant operations. Usually, inevitable model inaccuracies are dealt by collecting the plant measurements at the local operating conditions in order to adapt model parameters, followed by numerical re-optimization. This iterative two-step procedure often results in a sub-optimal solution, since the models are typically not designed for optimization.

Modifier Adaptation (MA) is a Real-Time Optimization (RTO) technique that directly adds the affine-correction terms to the model. The affine corrections are parametrized in modifiers that are tailored to the optimization needs. This enables modifier adaptation to guarantee, upon convergence, matching the plant and the modified model's optimality conditions. However, computing the modifiers requires estimates of the plant gradients that are obtained via expensive plant experiments. The experimental cost of gradient estimation can be reduced by relying more on the model of the considered plant. For example, Directional Modifier Adaptation (DMA) relies on offline-computed local parametric sensitivity analysis performed on the gradient of the Lagrangian function of the model resulting in reduced number of input directions that describe the gradient uncertainty in the model. Thereby, plant gradients are estimated only in a low-dimensional space of privileged input directions considerably reducing the experimental costs. However, local sensitivity analysis is often ineffective when the gradient of the model is considerably nonlinear in parameters.

This thesis proposes an online procedure based on global sensitivity analysis for finding the most promising privileged directions that adequately compensates for the model deficiencies in predicting the plant optimality conditions. The discovered privileged directions are such that, upon parametric perturbations, the gradient varies a lot along the privileged directions and varies only a little along the remaining input directions. Consequently, the gradients of the model cost and constraints are corrected only along the privileged directions by adapting modifiers. The resulting methodology is named as Active Directional Modifier Adaptation (ADMA). Several simulation studies conducted show that the proposed approach reaches the near-optimality conditions at a considerably reduced experimental cost.

In addition, this thesis attempts to establish a direct relation between the optimality conditions and the parameters of a given model. Model parameters are analyzed to

discover *mirror* parameters that mimic the behavior of modifiers in influencing the optimality conditions. It is proposed to adapt mirror parameters instead of modifiers yielding the benefit of both, modifier adaptation in enforcing optimality conditions and of parameter adaptation in better noise handling and convergence.

Moreover, it is also investigated how to establish the synergies between privileged input directions with model parameters in order to reduce experimental efforts. The steady-state optimization of a simulated chemical process shows that the privileged directions and the selected parameters work in perfect harmony to effectively reach near-optimal performance.

Finally, the study on the power maximization of flying kites leads to the development of trust-region based method to better control the input step size. It is shown that this approach improves performance significantly.

Keywords: Real-time optimization, Modifier adaptation, Parameter estimation, Active subspaces, Dimension reduction, Sensitivity analysis.

Résumé

Les procédés industriels sont mis en œuvre dans le but de maximiser le profit économique tout en respectant les contraintes critiques du processus. À cette fin, une optimisation basée sur un modèle peut être effectuée afin d'assurer un fonctionnement optimal du procédé. Habituellement, les inexactitudes inévitables du modèle sont traitées par la collecte des mesures du procédé dans les conditions d'exploitation locales afin d'adapter les paramètres du modèle, suivie d'une ré-optimisation numérique. Cette procédure itérative en deux étapes aboutit souvent à une solution sous-optimale, car les modèles ne sont généralement pas conçus pour l'optimisation.

Modifier Adaptation (MA) est une technique d'optimisation en temps réel (RTO) qui ajoute directement les termes affine-correction au modèle. Les corrections affines sont paramétrées dans des modifiers adaptés aux besoins d'optimisation. Cela permet d'adapter le modifier pour garantir, lors de la convergence, l'adéquation entre le procédé et les conditions optimales du modèle modifié. Cependant, le calcul des modifiers nécessite des estimations des gradients du procédé qui sont obtenus par des expérimentations coûteuses. Le coût expérimental de l'estimation du gradient peut être réduit en s'appuyant davantage sur le modèle du procédé considérée. Par exemple, l'adaptation des modifiers directionnels (DMA) repose sur une analyse de sensibilité paramétrique locale hors ligne effectuée sur le gradient de la fonction Lagrangienne du modèle, ce qui permet de réduire le nombre de directions d'entrée qui décrivent l'incertitude du gradient dans le modèle. Ainsi, les gradients du procédé ne sont estimés que dans un espace de faible dimension de directions d'entrée privilégiées, ce qui réduit considérablement les coûts expérimentaux. Cependant, l'analyse de sensibilité locale est souvent inefficace lorsque le gradient du modèle est considérablement non linéaire dans les paramètres.

Cette thèse examine les schémas RTO basés sur les outils de l'analyse de sensibilité globale. Ces outils permettent de trouver les directions privilégiées les plus prometteuses qui compensent adéquatement les déficiences du modèle dans la prédiction des conditions optimales du procédé. Les directions privilégiées découvertes sont telles que, lors de perturbations paramétriques, le gradient varie beaucoup le long des directions privilégiées et ne varie que peu le long des directions d'entrée restantes. Par conséquent, les gradients de coûts et de contraintes du modèle ne sont corrigés que dans les directions privilégiées en adaptant les modifiers. Plusieurs études de simulation réalisées montrent que l'approche proposée atteint des conditions quasi-optimales

à un coût expérimental considérablement réduit.

De plus, cette thèse tente d'établir une relation directe entre les conditions optimales et les paramètres d'un modèle donné. Les paramètres du modèle sont analysés pour découvrir les paramètres qui imitent le comportement des modifiers en influençant les conditions optimales. Il est proposé d'adapter les paramètres miroirs au lieu de modifiers, ce qui permet de bénéficier à la fois de l'adaptation des modifiers pour imposer des conditions optimales et de l'adaptation des paramètres pour améliorer la gestion du bruit et la convergence.

De plus, il est également étudié comment trouver des synergies entre les directions d'entrée privilégiées avec les paramètres du modèle afin de réduire les efforts expérimentaux. L'optimisation en régime permanent d'un procédé chimique simulé montre que les directions privilégiées et les paramètres sélectionnés travaillent en parfaite harmonie pour atteindre efficacement des performances quasi-optimales.

Enfin, l'étude sur la maximisation de la puissance des cerfs-volants conduit au développement d'une méthode basée sur les régions de confiance pour mieux contrôler la taille des pas d'entrée. Il est démontré que cette approche améliore significativement les performances.

Mots-clés : Optimisation en temps réel, Modifier adaptation, Estimation des paramètres, sous-espaces actifs, Réduction des dimensions, Analyse de sensibilité.

Contents

List of Figures

List of Tables

1 Introduction

1.1 Motivation

Economic profits of process plants are direct functions of their operating conditions. For large-scale plants, even 1% gain in the yield via operational improvements can lead to significant economic benefits [37]. Real-time optimization (RTO) attempts achieving continuous improvement in plant operations with the goal of reaching plant optimality, while simultaneously meeting constraints on process safety, equipment longevity, and product quality. RTO has resulted in tremendous economic benefits to various processes, ranging from production of petrochemicals to efficient fuel cell operations to polymerization processes to pharmaceutical manufacturing [20, 51, 78, 115].

RTO has gained further importance as today's dynamic markets demand for ever changing product specifications and, in addition, price fluctuations in the raw materials and the finished products demand for technological solutions that suitably respond in real-time. Moreover, continuous long-term usage of production equipment accelerates the degradation of the operating plant that adds to the challenge of reaching economically beneficial plant operations. RTO aims at addressing these issues by offering solutions that can respond quickly and efficiently, without loosing sight of the process limitations.

To this end, RTO typically deploys mathematical models derived from first principles and/or historical operating data. The RTO performance is significantly dependent on the quality of the models used [44, 133]. These models are traditionally designed with the goal of understanding the relationships among various plant variables and explain the behavior of the plant. This has to be achieved despite the fact that the accuracy of the model predictions are highly dependent on the quality of the data used, the depth of the modeling process, quality of the design of experiments, parameter selection process and the model validation procedure [11, 54, 55, 66, 72, 123]. Moreover, these models are not tailored to meet RTO-specific requirements of accurate predic-

tion of the plant optimality conditions [22]. Therefore, the traditional RTO methods that adapt the model parameters may well fail to reach plant optimality [44, 82]. This motivates towards understanding the relationship between model parameters and the optimality conditions. Such analysis should be targeted to establish how the parameter adaptation directly affects the optimality conditions and to what degree. Surprisingly, the current RTO literature is quite limited in this regard [22, 80].

One way to compensate for the lack of model accuracy is by relying more on information-rich plant data. The collected data is translated into a few correction terms that are added to the concerned model. These correction terms are tailored to offset the model discrepancy in predicting the plant optimality conditions [16, 21, 33, 56, 58, 83, 101, 103]. The RTO techniques that utilize such mechanism involve local estimation of the plant quantities that characterize the optimality conditions. To this end, tailored experiments are performed to gather the relevant plant estimates. These estimates are then utilized for local plant-model matching that moves the plant towards its optimal state via (corrected) model-based optimization. Such methods have shown promising results in practice [20, 31, 87, 95, 135].

The added correction terms are parameterized by *modifiers* that are iteratively adapted to achieve plant-model matching. Although, the adaptation of modifiers guarantees plant optimality [12, 17, 83, 85], it often demands for excessive plant experiments for estimating the requisite quantities [31, 33]. The experimental process does not only consume resources, but also slows down the whole optimization process. If the change in plant conditions is at higher frequencies than the optimization process, one may never reach plant optimality.

An alternative to reduce the reliance on excessive plant experiments is to understand the dependence of the optimality conditions on the model parameters [33]. Such analysis is done to check if plant- model mismatch can be compensated by only partial corrections and thereby, less experiments would be performed to gather only the most relevant plant information.

Motivated by above, the work in this thesis is driven by two main goals—(a) To understand the dependence of the optimality conditions on the model parameters, thereby, design minimal number of plant experiments that can best compensate for the model deficiencies in predicting plant optimality. (b) To establish a direct relationship between optimality conditions and model parameters that results in relevant pairing between different elements of the optimality conditions and the model parameters. This way, a set of parameters could be discovered that act similar to modifiers in compensating for plant-model mismatch. To meet the two goals, tools of global sensitivity analysis are borrowed from the literature and tailored to achieve the desired results.

1.2 State of the Art

RTO can be classified as either implicit or explicit RTO [52]. The two classes of RTO are described below.

1.2.1 Implicit RTO

This class of RTO methods proposes to select outputs whose optimal values are approximately invariant to uncertainty. The main idea is to keep these outputs close to their invariant values and thus attenuate the effect of uncertainty by adapting the inputs directly on the basis of measurements. In other words, a feedback law is sought that implicitly solves the optimization problem. One of such schemes called NCO Tracking [125, 126, 127] has the attractive feature that it can handle constrained problems, although this requires that the active set remains unchanged in the presence of uncertainty. Another scheme called Neighboring-Extremal Control (NEC) [50, 61] combines a variational analysis of the model at hand with output measurements to enforce the plant Necessary Conditions of Optimality (NCO)s [1]. With Extremum-Seeking Control (ESC), dither signals are added to the inputs such that an estimate of the plant cost gradient is obtained online using output measurements [6, 62, 73]. In Self-Optimizing Control (SOC) [3, 67, 68, 69, 121], a linear or a polynomial combination of the plant outputs is selected that is invariant to disturbances. The chosen combination is then kept at constant setpoint values in closed loop to minimize optimality loss.

The methods described above belong to the class of implicit RTO methods (see Figure 1.1) as they do not adapt the models utilized to select the invariant output variables, therefore, do not repeat the numerical optimization explicitly. Such methods directly update the inputs via feedback control.

1.2.2 Explicit RTO

Explicit RTO methods perform a measurement-based adaptation of either the model parameters and/or the correction terms that are added to the model. The updated models are then utilized in the optimization layer to compute the inputs [15, 16, 24, 83]. Some of the explicit-RTO schemes are discussed next.

Since accurate models are rarely available in industrial applications, RTO typically proceeds by an iterative two-step approach [24, 65, 89] (see Figure 1.2a), namely a model update step followed by an optimization step. The model-update step typically consists of a parameter estimation problem. The objective is to find values of selected model parameters for which the model gives a good prediction of the measured plant outputs at the local input values. In the optimization step, the updated model is used

[1] The main focus in this thesis is to enforce the Karush-Kuhn-Tucker (KKT) conditions [10].

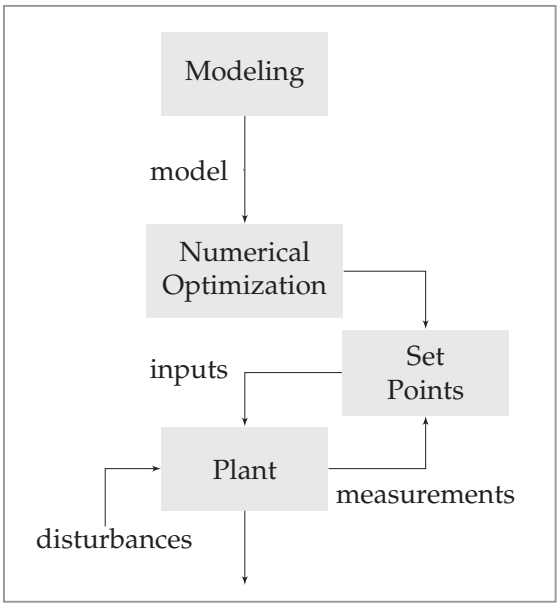


Figure 1.1 – Implicit RTO schemes.

to determine a new input by solving a model-based optimization problem.

The classical two-step approach works well provided that (i) there is little structural plant-model mismatch [133], and (ii) the changing operating conditions provide sufficient excitation for estimating the uncertain model parameters. Unfortunately, such conditions are rarely met in practice. Regarding (i), in the presence of structural plant-model mismatch, it is typically not possible to satisfy the plants NCOs simply by estimating the model parameters that predict the plant outputs well. Since the NCOs are characterized by gradients, therefore, parameters need to be adjusted such that the plant and the model outputs have matching gradients. Recently, an RTO methodology called Simultaneous Model Identification and Optimization (SMIO), that adapts the model parameters in order to match the outputs and the gradients of the cost and the constraint functions of the optimization problem, has been proposed in [79, 80]. SMIO also proposes to add correction terms as the parameter adaptation alone is insufficient in matching all the targeted quantities.

Another technique that utilizes correction terms is referred to as Integrated System Optimization and Parameter Estimation (ISOPE) [16, 101, 102, 103]. ISOPE requires both output measurements and estimates of the gradients of the plant outputs with respect to the inputs. These gradients allow computing the plant cost gradient that is used to modify the cost function of the model-based optimization problem. The matching of the model cost gradient and the plant cost gradient facilitates the model-based optimization problem to better approximate the plant optimization problem. With ISOPE, process measurements are incorporated at two levels, namely, the model parameters are adapted via output measurements, and the cost function is corrected

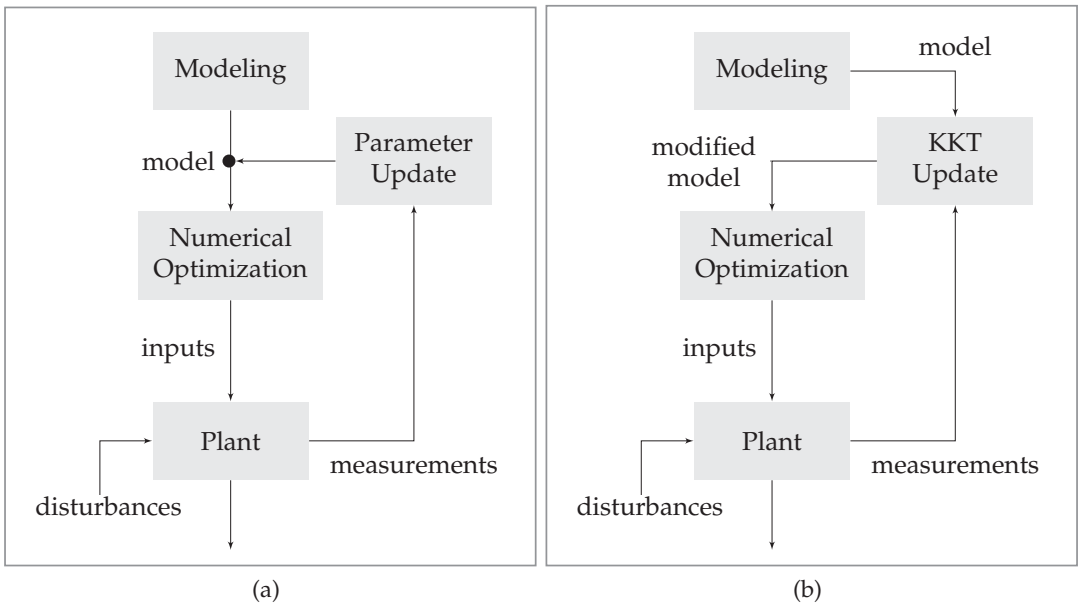


Figure 1.2 – Explicit RTO schemes. (a) Two-step approach. (b) Modifier adaptation.

by the addition of an affine term in inputs. The gradient correction constitutes the affine term.

In contrast to ISOPE, Modifier Adaptation (MA) (see Figure 1.2b) keeps the model parameters fixed, and affine correction terms (parametrized in modifiers) are added to the model cost and constraint functions [83, 86]. Modifiers are adapted iteratively to locally match the plant NCOs. Each element of the modifiers can manipulate only a single component of either the model constraint or the cost/constraint gradients without affecting the other components. This enables MA schemes to guarantee KKT matching [12, 17, 83, 86]. The modifiers are tailored to meet the NCOs and, thereby, provide the flexibility that the parameters of the model may lack [22].

For the implementation of MA, plant measurements are expected to be sufficiently rich to allow good estimates of the plant cost and constraint values and of their gradients. The most straightforward way to estimate the gradients is by finite differences, which requires perturbing the inputs around their nominal values and collecting the output measurements after each plant experiment.

In the past years, several methods have been proposed to obtain the gradient information. In dual MA [84], one considers an additional constraint in the RTO problem, which restricts the location of the next RTO input such that reliable gradient information can be extracted using the current input and the previously visited RTO inputs. Dual ISOPE [16] and the approach proposed in [104] also make use of 'duality constraints' so as to simultaneously optimize and estimate the gradients. Alternatively, it

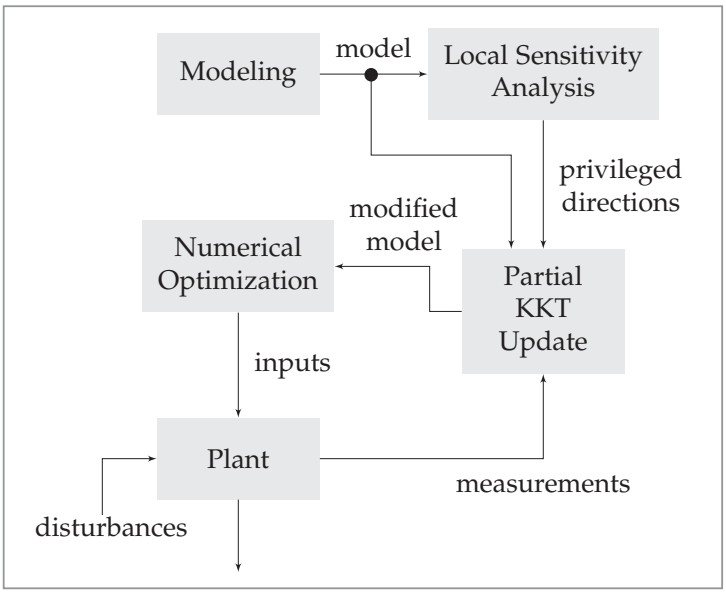


Figure 1.3 – Directional modifier adaptation.

is proposed to combine data-based quadratic approximation and MA to obtain accurate gradient estimates [58]. Instead of finding gradients, one can attempt to directly compute the first-order correction terms using an additional optimization layer as proposed by [93]. In addition, it has been proposed to use the transient measurements for gradient estimation in [38, 47].

The gradient estimation by perturbing the plant is resource consuming and the experimental cost increases with increasing input dimensions. To reduce the experimental burden of gradient estimation, one can rely more on the process models. With this philosophy, an MA-based approach has been proposed in [33] that lowers the expense of gradient estimation by questioning the necessity of correcting the full gradients. This approach, labeled Directional Modifier Adaptation (DMA) [33] (see Figure 1.3), proposes to correct the model gradients only in *privileged* input directions that span a reduced subspace of the input space. This subspace is computed only once offline by means of a local sensitivity analysis conducted on the gradient of the Lagrangian function of the model. Hence, DMA results in only partial corrections of the gradients that characterize the KKT conditions, thereby resulting in only partial KKT matching. It is assumed that the given model is able to accurately predict the remaining plant quantities that symbolize the KKT conditions.

In the RTO literature, several variants of modifier adaption exist. One of them is that of Constraint Adaptation (CA) [21]. In CA, only constraint functions are corrected to the zeroth order and gradient corrections to the cost and the constraints are not made. This scheme performs well when the model gradients are accurate enough to push towards the correct set of active constraints that characterize the plant optimality.

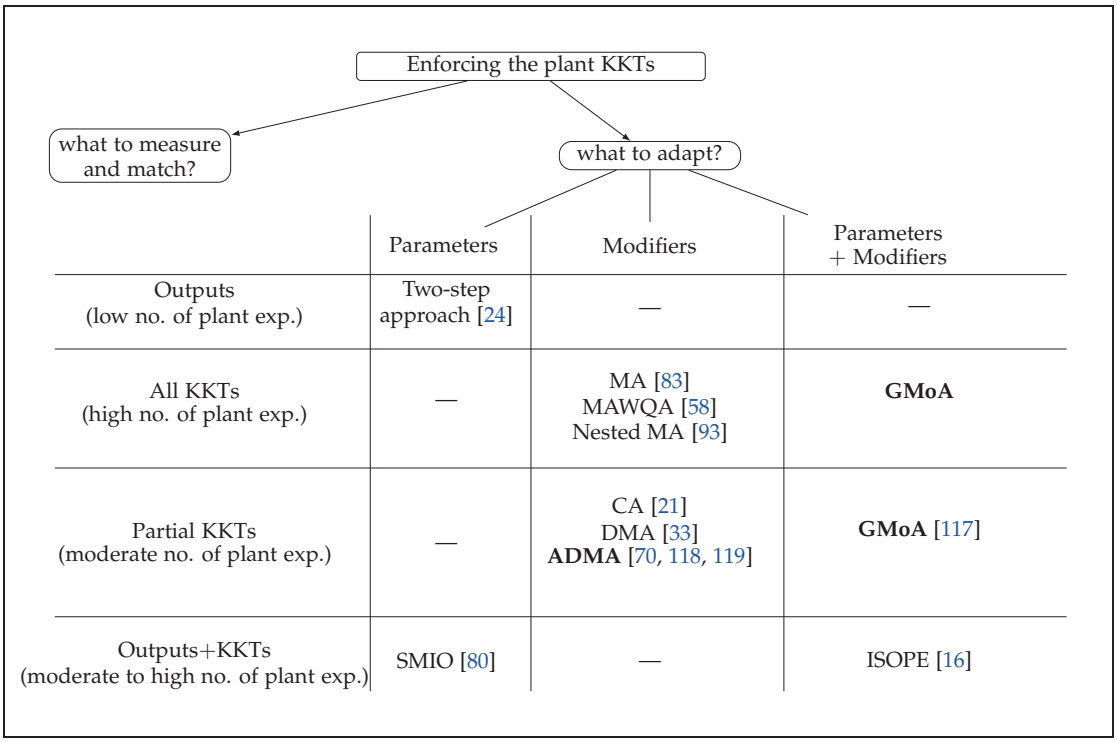


Figure 1.4 – Classification of explicit RTO schemes. **Bold:** RTO methodologies proposed in the thesis.

Other MA variant in [48] include the use of convex models to address the optimality conditions concerning the Hessian information. An important issue of feasible-side convergence of MA-based schemes is addressed in [19, 85, 88, 116, 120]. The handling of measurement noise is discussed in [18, 84]. Notably, the application of derivative-free methods [27, 96] for attenuating the impact of noise has shown good results in the MA-based scheme called MAWQA [58]. Recently, MA has also been extended to distributed RTO problems [90, 114].

Figure 1.4 classifies some of the explict RTO schemes. An RTO scheme is categorized based on whether it attempts to match the plant outputs or the pant quantities that characterize the KKT conditions, namely, the plant cost and constraints gradients and the zeroth-order constraint values. Some RTO schemes like SMIO attempt to match both, the outputs and the KKT conditions. In addition, each RTO scheme is also distinguished based on whether it adapts the model parameters or the modifiers or both. The number of plant experiments required by each scheme depends on whether it attempts to match all or only selected plant quantities to enforce optimality. An ideal scheme should be able to balance the experimental costs without compromising too much on plant optimality. This thesis proposes two different explicit RTO methodologies highlighted in bold in the Figure 1.4.

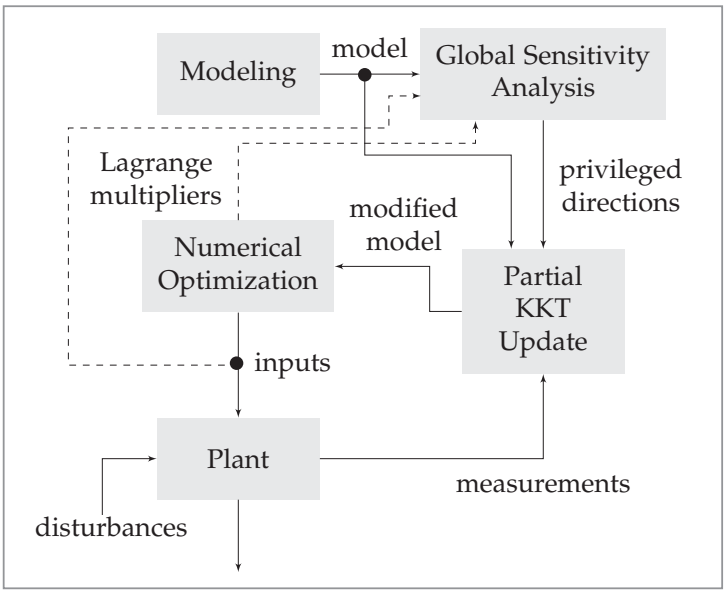


Figure 1.5 – The first proposed approach - Active directional modifier adaptation.

1.3 Contributions of the Thesis

This thesis propose methods which attempt to extract the most out of a given model in predicting plant optimality. To this end, a tool for Global Sensitivity Analysis (GSA), known as active subspaces [28], is utilized and altered. In active subspaces, parametric perturbations are performed to discover a low-dimensional structure in parameter space that is most responsible for the variations of the quantity of interest. This low-dimensional structure represents the *influential* parameter space.

The chapter-wise thesis contributions are as follows:

Chapter 3

- The building blocks of active subspace theory are tailored to discover a low-dimensional structure in input space (called privileged directions) rather than in parameter space.

- An Active Directional Modifier Adaptation (ADMA, see Figure 1.5) is proposed that guarantees plant optimality upon convergence under parametric plant-model mismatch. In the presence of structural plant-model mismatch, ADMA guarantees optimality limited to the privileged directions space.

- Two different simulation studies of run-to-run optimization of the semi-batch rectors confirm the presence of low-dimensional privileged directions that adequately compensate for model deficiencies in reaching plant optimality. This

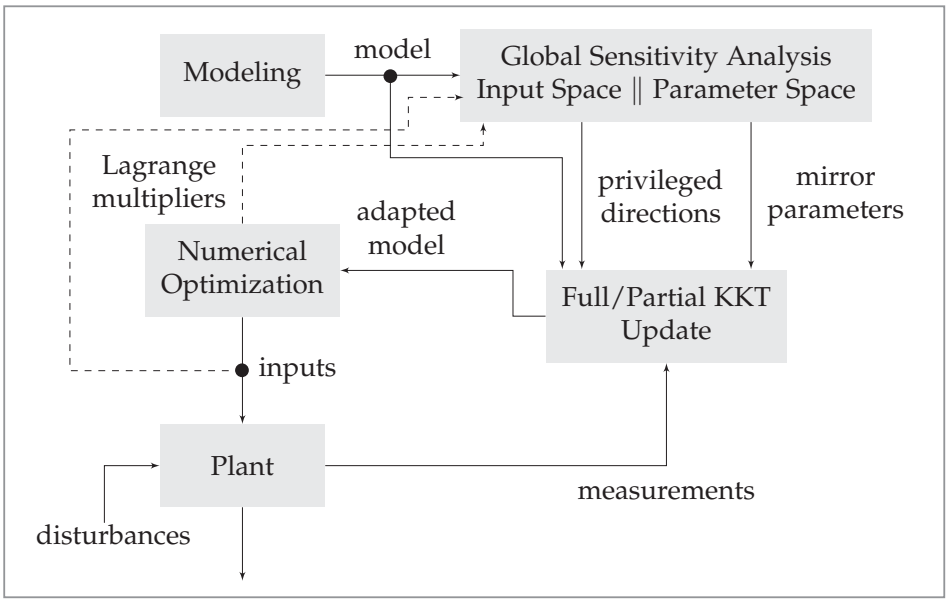


Figure 1.6 – The second proposed approach - Generalized model adaptation.

way, the experimental cost is reduced drastically.

- The active subspace tools are further tailored and the parametric sensitivities are computed indirectly, which considerably reduces the computational cost of global sensitivity analysis. Alternatively, when the parameter dimensions are large, active subspaces are directly utilized to reduce the computational costs.

- The chapter resulted in the publications [118, 119]. Another peer-reviewed journal submission is pending regarding the computational cost of global sensitivity analysis.

Chapter 4

- A Generalized Model Adaptation (GMoA) framework (see Figure 1.6) is proposed that adapts a set of model parameters called mirror parameters. Mirror parameters mimic the modifier behavior and, therefore, their adaptation has advantages of both, modifier and parameter adaptation. With the help of modifiers, the framework can be adapted to perform either full or only selected KKT matching. Mirror parameters are found by direct use of active subspaces.

- Gradients in the privileged input directions are matched by adapting corresponding influential parameters instead of modifiers, thereby maximizing the model utilization for accurate prediction of plant optimality at a reasonable experimental cost.

- This chapter resulted in the publication [117]. Another peer-reviewed journal publication on GMoA framework is pending for submission.

Chapter 5

- Simulation study of power maximization in flying kites leads to the development of trust-region-based ADMA that effectively controls the input step lengths to ensure optimality.
- The second simulation study of fuel-cell system compares various MA-based schemes. It is shown that ADMA finds a nice balance between plant optimality and experimental cost. Moreover, in ADMA, active subspaces are utilized to reduce the computational costs of obtaining privileged directions from one hour to only 32 seconds.
- This chapter resulted in the publication [70]. Another peer-reviewed journal publication on fuel cell simulation is pending for submission.

1.4 Organization of the Thesis

The next chapter is on preliminaries that are the building blocks of the techniques discussed and developed in the later chapters. These include the RTO problem definition, various conditions that define the optimal solution, two-step approach, MA schemes and their properties, active subspaces and sufficient summary plots.

Chapter 3 designs the global sensitivity analysis for discovering privileged input directions that are utilized to develop the ADMA methodology. ADMA is illustrated on two different simulation studies. The chapter further proposes two different alternatives to reduce the computational expense of finding privileged directions via global sensitivity analysis.

Chapter 4 develops the framework of generalized model adaptation by introducing the concept of mirror parameters that, similar to modifiers, enable independent matching of the KKT conditions. The chapter further discusses the synergies between the parameters and modifiers and how both can be combined to enable independent KKT matching. In addition, the chapter proposes to pair the privileged directions with corresponding mirror parameters to reach plant optimality at reasonable experimental costs. The proposed concepts are illustrated on a simulation study.

Chapter 5 discusses the application of ADMA on two advanced simulation studies of power maximization of flying kites and maximizing efficiency of a fuel-cell system. A trust-region framework is designed for step-length control in kites. Different MA-based techniques are compared on the fuel cell study.

Finally, chapter 6 concludes the thesis and provides perspectives on future research directions.

2 Preliminaries

In this thesis, two types of RTO problems are dealt with, steady-state optimization of continuously operating processes and run-to-run optimization of semi-batch or batch operations. The run-to-run optimization problem is converted to a static-optimization problem upon input parametrization. The static optimization is formulated as a nonlinear programming problem. The RTO problem formulation and related optimality conditions are described in Sections 2.1 and 2.2, respectively. The classical two-step approach is discussed next in Section 2.3. In Subsection 2.3.2, the important concept of model adequacy is described. This subsection further discusses the conditions for the two-step approach to be model adequate. Section 2.4 discusses the bias-corrections-based constraint adaptation and its properties that include guaranteed plant constraint satisfaction upon convergence. Section 2.5 describes modifier adaptation that additionally corrects the model gradients. Modifier adaptation's various features including guranteed plant optimality upon convergence, model adequacy conditions and its Lagrangian-modifier form are detailed in Subsections 2.5.2 and 2.5.3. Section 2.6 discusses the concept of influential and non-influential parameters that can be found via various sensitivity analysis techniques. This thesis heavily relies on the global sensitivity analysis technique of active subspaces that is detailed in Subsection 2.6.1. Subsection 2.6.2 recaps the Monte-Carlo sampling-based algorithm for computing active subspaces and, finally, Subsection 2.6.3 describes the plotting tools called sufficient summary plots that help visually identify the presence of a one- or two-dimensional active subspace.

2.1 Problem Formulation

The optimization problem for the plant reads

$$\min_{\mathbf{u}} \quad \Phi_p(\mathbf{u}) := \phi(\mathbf{u}, \mathbf{y}_p(\mathbf{u})) \tag{2.1a}$$

$$\text{s.t.} \quad \mathrm{G}_{p,i}(\mathbf{u}) := \mathrm{g}_i(\mathbf{u}, \mathbf{y}_p(\mathbf{u})) \leq 0, \quad i = 1, \ldots, n_{\mathrm{g}}, \tag{2.1b}$$

where $\mathbf{u} \in \mathbb{R}^{n_{\mathrm{u}}}$ is the vector of input variables, $\mathbf{y}_p \in \mathbb{R}^{n_{\mathrm{y}}}$ are the measured output variables, $\phi : \mathbb{R}^{n_{\mathrm{u}}} \times \mathbb{R}^{n_{\mathrm{y}}} \to \mathbb{R}$ is the cost to be minimized, $\mathrm{g}_i \in \mathbb{R}$, $i = 1, \ldots, n_{\mathrm{g}}$, are the inequality constraints. The optimal input vector of Problem (2.1) is denoted as $\mathbf{u}_p^\star$.

The main challenge in solving the above optimization problem stems from the fact that the (steady-state/static) input-output mapping $\mathbf{y}_p(\mathbf{u})$ is unknown. However, an approximate process model is assumed to be available, with the input-output mapping $\mathbf{y}(\mathbf{u}, \boldsymbol{\theta})$, where $\boldsymbol{\theta} \in \mathbb{R}^{n_\theta}$ are the model parameters. Using the model, Problem (2.1) can be approximated as

$$\min_{\mathbf{u}} \Phi(\mathbf{u}, \boldsymbol{\theta}) := \phi(\mathbf{u}, \mathbf{y}(\mathbf{u}, \boldsymbol{\theta})) \tag{2.2a}$$

$$\text{s.t.} \quad \mathrm{G}_i(\mathbf{u}, \boldsymbol{\theta}) := \mathrm{g}_i(\mathbf{u}, \mathbf{y}(\mathbf{u}, \boldsymbol{\theta})) \leq 0, \quad i = 1, \ldots, n_{\mathrm{g}}. \tag{2.2b}$$

The nominal optimal input $\mathbf{u}^\star$ is found by solving Problem (2.2) for a fixed value $\boldsymbol{\theta} = \boldsymbol{\theta}_0$, where $\boldsymbol{\theta}_0$ is the vector of nominal model parameters. In the presence of plant-model mismatch, the model optimum $\mathbf{u}^\star$ may not be equal to the plant optimum $\mathbf{u}_p^\star$. The goal of RTO is to find $\mathbf{u}_p^\star$ by iteratively adapting (modifying) and solving Problem (2.2).

This thesis focuses on both, iterative steady-state optimization of continuous processes and run-to-run optimization of batch/semi-batch processes. For the run-to-run optimization, the dynamic optimization problems are reformulated as static optimization problems by finite discretization of the inputs, constraints and the dynamic models similar to [35, 49]. The resulting static models are then adapted iteratively, improving the batch/semi-batch performance from one run to the next.

2.2 Necessary Conditions of Optimality

Necessary conditions of optimality (NCO) characterize the local minimum of an optimization problem such as (2.2). The NCOs depend on the set of active constraints, which is defined as follows:

$$\mathcal{A}(\mathbf{u}, \boldsymbol{\theta}) = \left\{ i \in \{1, \ldots, n_{\mathrm{g}}\} \,|\, \mathrm{G}_i(\mathbf{u}, \boldsymbol{\theta}) = 0 \right\}. \tag{2.3}$$

The so-called Linear Independence Constraint Qualification (LICQ) requires that the gradients of the active constraints be linearly independent. If at a local minimum $\mathbf{u}^\star$, the model cost and constraint functions Φ and $\mathbf{G}$ are differentiable and LICQ holds, then there exists unique value of the Lagrange multiplier vector $\boldsymbol{\mu}^\star$ such that the following KKT conditions are satisfied at $(\mathbf{u}^\star, \boldsymbol{\mu}^\star)$ [10],

$$\begin{aligned} &\mathbf{G} \leq \mathbf{0}, \quad \boldsymbol{\mu}^\mathsf{T}\mathbf{G} = 0, \quad \boldsymbol{\mu} \geq \mathbf{0}, \\ &\frac{\partial \mathcal{L}}{\partial \mathbf{u}} = \frac{\partial \Phi}{\partial \mathbf{u}} + \boldsymbol{\mu}^\mathsf{T}\frac{\partial \mathbf{G}}{\partial \mathbf{u}} = \mathbf{0}, \end{aligned} \tag{2.4}$$

where $\mathcal{L}(\mathbf{u}, \boldsymbol{\mu}, \boldsymbol{\theta}) := \Phi(\mathbf{u}, \boldsymbol{\theta}) + \boldsymbol{\mu}^\mathsf{T}\mathbf{G}(\mathbf{u}, \boldsymbol{\theta})$ is the Lagrangian function of the model. A solution $\mathbf{u}^\star$ satisfying these conditions is called a KKT point.

If LICQ holds at $\mathbf{u}^\star$, then one can write,

$$\frac{\partial \mathbf{G}^\mathsf{a}}{\partial \mathbf{u}}(\mathbf{u}^\star, \boldsymbol{\theta})\boldsymbol{Z} = \mathbf{0}, \quad \mathbf{G}^\mathsf{a}(\mathbf{u}^\star, \boldsymbol{\theta}) \in \mathbb{R}^{n_\mathrm{g}^\mathsf{a}},$$

where $\mathbf{G}^\mathsf{a}(\mathbf{u}^\star, \boldsymbol{\theta})$ is the vector of active constraints at $\mathbf{u}^\star$ and n_g^a is the cardinality of $\mathcal{A}(\mathbf{u}^\star, \boldsymbol{\theta})$; and $\boldsymbol{Z} \in \mathbb{R}^{n_\mathrm{u} \times (n_\mathrm{u} - n_\mathrm{g}^\mathsf{a})}$ is the null-space matrix. The reduced Hessian of the Lagrangian on this null space, $\nabla_r^2 \mathcal{L}(\mathbf{u}^\star, \boldsymbol{\theta}) \in \mathbb{R}^{(n_\mathrm{u} - n_\mathrm{g}^\mathsf{a}) \times (n_\mathrm{u} - n_\mathrm{g}^\mathsf{a})}$, is described in [59] as:

$$\nabla_r^2 \mathcal{L}(\mathbf{u}^\star, \boldsymbol{\theta}) := \boldsymbol{Z}^\mathsf{T} \left[\frac{\partial^2 \mathcal{L}}{\partial \mathbf{u}^2}(\mathbf{u}^\star, \boldsymbol{\mu}^\star, \boldsymbol{\theta}) \right] \boldsymbol{Z}.$$

In addition to the first-order KKT conditions, a second-order necessary condition for a local minimum is the requirement that $\nabla_r^2 \mathcal{L}(\mathbf{u}^\star, \boldsymbol{\theta})$ be positive semi-definite. On the other hand, $\nabla_r^2 \mathcal{L}(\mathbf{u}^\star, \boldsymbol{\theta})$ being positive definite is sufficient for a strict local minimum [59].

2.3 Two-Step Approach

2.3.1 The Concept

The classical two-step approaches in RTO propose to adapt model parameters with the aim of matching the plant and the model outputs [24, 89, 103]. The updated parameters are then plugged into the model-based optimization problem in (2.2), thus resulting in the new input value. The plant outputs $\mathbf{y}_p$ corresponding to the updated inputs are measured, and new measurements are utilized to redo the parameter estimation step. This iterative procedure is repeated until convergence. The two main steps of the approach are the parameter estimation step and the optimization step that take the following form at the k^{th} RTO iteration,

Parameter estimation step :

$$\boldsymbol{\theta}_{k+1} = \arg\min_{\boldsymbol{\theta}} \|\mathbf{y}_p(\mathbf{u}_k) - \mathbf{y}(\mathbf{u}_k, \boldsymbol{\theta})\|^2, \tag{2.5}$$

where $\|\cdot\|$ is the Euclidean norm; $\mathbf{y}_p(\mathbf{u}_k)$ are the plant measurements at the current input value $\mathbf{u}_k$. The updated model parameters are then used in the optimization problem (2.2) to generate the new input value $\mathbf{u}_{k+1}$ as follows:

Optimization step :

$$\mathbf{u}_{k+1} = \min_{\mathbf{u}} \Phi(\mathbf{u}, \boldsymbol{\theta}_{k+1}), \tag{2.6a}$$

$$\text{s.t.} \quad \mathrm{G}_i(\mathbf{u}, \boldsymbol{\theta}_{k+1}) \leq 0, \quad i = 1, \ldots, n_{\mathrm{g}}. \tag{2.6b}$$

2.3.2 Model-Adequacy Criterion

Selection of the process model for RTO is a challenging task, as the quality of the solution obtained in (2.2) directly depends on the model employed [43, 44, 45]. In the context of RTO, a model is adequate if upon certain measurement-based adjustments (such as parameter adaptation), it is able to produce the plant optimum $\mathbf{u}_p^\star$ by solving the Problem (2.2). Model adequacy is formally defined as follows:

Definition 2.1 (Model-adequacy criterion [45])**.** *A process model is said to be adequate for use in an RTO scheme if it is capable of producing a fixed point that is a local minimum for the RTO problem at the plant optimum* $\mathbf{u}_p^\star$*.*

Therefore, when the two-step approach is employed as an RTO scheme, the model used must have the parameter value $\boldsymbol{\theta}^\star$, such that $\mathbf{u}_p^\star$ is a fixed point of the Problem (2.2) for $\boldsymbol{\theta} = \boldsymbol{\theta}^\star$. For the two-step approach, the following model-adequacy conditions are proposed.

Criterion 2.1 (Model adequacy for the two-step approach [45])**.** *Let* $\mathbf{u}_p^\star$ *be the unique plant optimum and parameter values* $\boldsymbol{\theta}^\star$ *exist such that*

$$\mathrm{G}_i(\mathbf{u}_p^\star, \boldsymbol{\theta}^\star) = 0, \quad i \in \mathcal{A}(\mathbf{u}_p^\star, \boldsymbol{\theta}^\star), \tag{2.7a}$$

$$\mathrm{G}_i(\mathbf{u}_p^\star, \boldsymbol{\theta}^\star) < 0, \quad i \notin \mathcal{A}(\mathbf{u}_p^\star, \boldsymbol{\theta}^\star), \tag{2.7b}$$

$$\nabla_r \mathcal{L}(\mathbf{u}_p^\star, \boldsymbol{\theta}^\star) = \mathbf{0}, \tag{2.7c}$$

$$\nabla_r^2 \mathcal{L}(\mathbf{u}_p^\star, \boldsymbol{\theta}^\star) > \mathbf{0}, \tag{2.7d}$$

$$\frac{\partial J^{id}}{\partial \boldsymbol{\theta}}(\mathbf{y}_p(\mathbf{u}_p^\star), \mathbf{y}(\mathbf{u}_p^\star, \boldsymbol{\theta}^\star)) = \mathbf{0}, \tag{2.7e}$$

$$\frac{\partial^2 J^{id}}{\partial \boldsymbol{\theta}^2}(\mathbf{y}_p(\mathbf{u}_p^\star), \mathbf{y}(\mathbf{u}_p^\star, \boldsymbol{\theta}^\star)) > \mathbf{0}, \tag{2.7f}$$

where $\nabla_r \mathcal{L}$ and $\nabla_r^2 \mathcal{L}$ are the reduced gradient and reduced Hessian of the model Lagrangian, respectively; and J^{id} denotes the objective function of the parameter estimation step in (2.5). Then, the model is said to be adequate for the two-step approach given in Section 2.3.1.

Model adequacy for the two-step approach has been studied in details in the literature [43, 45, 82]. Note that the plant optimum $\mathbf{u}_p^\star$ is not known a priori and, therefore, verifying model adequacy is not straightforward. Moreover, Marchetti in [82] showed, through simulation study of the Williams-Otto reactor [130], that even if a given process model satisfies the model-adequacy criteria, the two-step approach is incapable of reaching $\mathbf{u}_p^\star$.

2.4 Constraint Adaptation

Certain plants are such that their optimal performance is mostly driven by the set of active constraints. In these cases, one may reach near optimal performance simply by tracking constraints [20, 22, 117]. For this, an RTO approach known as constraint adaptation (CA) can be used [21]. CA is an iterative scheme that solves the following optimization problem to reach the plant optimum $\mathbf{u}_p^\star$:

$$\min_{\mathbf{u}} \quad \Phi_{m,k}(\mathbf{u}) := \Phi(\mathbf{u}, \boldsymbol{\theta}_0) \tag{2.8a}$$

$$\text{s.t.} \quad \mathbf{G}_{m,k}(\mathbf{u}) := \mathbf{G}(\mathbf{u}, \boldsymbol{\theta}_0) + \boldsymbol{\varepsilon}_k^{\text{G}} \leq \mathbf{0}, \tag{2.8b}$$

where $\boldsymbol{\varepsilon}_k^{\text{G}} \in \mathbb{R}^{n_g}$ is the vector of zeroth-order modifier with $\varepsilon_{i,k}^{\text{G}}$ as its i^{th} component; and $\mathbf{G} \in \mathbb{R}^{n_g}$ is the vector of model constraints G_i, $i = 1, \ldots, n_g$. At the k^{th} RTO iteration, the modifiers are computed as follows:

$$\boldsymbol{\varepsilon}_k^{\text{G}} = \mathbf{G}_p(\mathbf{u}_k) - \mathbf{G}(\mathbf{u}_k, \boldsymbol{\theta}_0), \tag{2.9}$$

where $\mathbf{G}_p \in \mathbb{R}^{n_g}$ is the vector of plant constraints. Note that parameter adaptation is not required in CA as the modifiers introduce bias corrections at each iteration, which suffices to track the plant constraints. Hence, model parameters are fixed at their nominal values $\boldsymbol{\theta}_0$.

CA can yield optimality without requiring estimation of plant gradients, which makes this scheme very attractive for practical applications [20]. However, many process optimization problems do require gradient information to reach plant optimality. Therefore, in such cases, modifier adaptation (MA) that additionally corrects model gradients becomes more attractive RTO scheme. Nonetheless, upon convergence, CA guarantees finding a feasible input value for the plant constraints.

2.5 Modifier Adaptation

Modifier adaptation introduces first-order correction terms that are added to the cost and constraint functions predicted by the nominal model. At the k^{th} RTO iteration, the next RTO inputs are computed by solving the following *modified* optimization problem:

$$\min_{\mathbf{u}} \quad \Phi_{m,k}(\mathbf{u}) := \Phi(\mathbf{u}, \boldsymbol{\theta}_0) + (\boldsymbol{\lambda}_k^{\Phi})^{\mathsf{T}}\mathbf{u} \tag{2.10a}$$

$$\text{s.t.} \quad \mathbf{G}_{m,k}(\mathbf{u}) := \mathbf{G}(\mathbf{u}, \boldsymbol{\theta}_0) + \boldsymbol{\varepsilon}_k^{\mathrm{G}} + (\boldsymbol{\lambda}_k^{\mathrm{G}})^{\mathsf{T}}(\mathbf{u} - \mathbf{u}_k) \leq \mathbf{0}, \tag{2.10b}$$

where $\mathbf{G} \in \mathbb{R}^{n_{\mathrm{g}}}$ is the vector of constraints G_i, $\boldsymbol{\varepsilon}_k^{\mathrm{G}} \in \mathbb{R}^{n_{\mathrm{g}}}$ are the zeroth-order modifiers for the constraints; $\boldsymbol{\lambda}_k^{\Phi} \in \mathbb{R}^{n_{\mathrm{u}}}$ and $\boldsymbol{\lambda}_k^{\mathrm{G}} \in \mathbb{R}^{n_{\mathrm{u}} \times n_{\mathrm{g}}}$ are the first-order modifiers for the cost and constraint functions, respectively. At RTO iteration k, the modifiers are computed as follows:

$$\boldsymbol{\varepsilon}_k^{\mathrm{G}} = \mathbf{G}_p(\mathbf{u}_k) - \mathbf{G}(\mathbf{u}_k, \boldsymbol{\theta}_0), \tag{2.11a}$$

$$(\boldsymbol{\lambda}_k^{\Phi})^{\mathsf{T}} = \nabla_{\mathbf{u}}\Phi_p(\mathbf{u}_k) - \nabla_{\mathbf{u}}\Phi(\mathbf{u}_k, \boldsymbol{\theta}_0), \tag{2.11b}$$

$$(\boldsymbol{\lambda}_k^{\mathrm{G}})^{\mathsf{T}} = \nabla_{\mathbf{u}}\mathbf{G}_p(\mathbf{u}_k) - \nabla_{\mathbf{u}}\mathbf{G}(\mathbf{u}_k, \boldsymbol{\theta}_0), \tag{2.11c}$$

where $\nabla_{\mathbf{u}}(\cdot)$ is the gradient of a scalar quantity or the Jacobian of a vector quantity with respect to $\mathbf{u}$. MA guarantees meeting the plant KKT conditions (Problem 2.1) upon convergence [83]. Gradient adaptation via first-order modifiers plays a key role in meeting the plant KKT conditions. However, finding reliable plant gradients is a costly task as it requires additional plant evaluations. If, for instance, the forward finite-difference approach is used, then the number of plant evaluations at each RTO iteration increases linearly with the dimension of the input space.

2.5.1 Filters for MA

MA locally corrects the zeroth-order and first-oder terms of the model. However, in the absence of higher-order correction terms, the local corrections may yield excessively large input steps. Such steps lead to oscillatory behavior when MA is applied to a plant. To avoid excessive corrections, Marchetti et al. in [83] propose to either filter the modifier terms or the input updates. The first-order filters on the modifiers are used as follows

$$\boldsymbol{\varepsilon}_{k+1}^{\mathrm{G}} = (\boldsymbol{I}_{n_{\mathrm{g}}} - \boldsymbol{K}^{\boldsymbol{\varepsilon}})\boldsymbol{\varepsilon}_k^{\mathrm{G}} + \boldsymbol{K}^{\boldsymbol{\varepsilon}}\big(\mathbf{G}_p(\mathbf{u}_{k+1}) - \mathbf{G}(\mathbf{u}_{k+1}, \boldsymbol{\theta}_0)\big), \tag{2.12}$$

$$\boldsymbol{\lambda}_{k+1}^{\Phi} = (\boldsymbol{I}_{n_{\mathrm{u}}} - \boldsymbol{K}^{\Phi})\boldsymbol{\lambda}_k^{\Phi} + \boldsymbol{K}^{\Phi}\big(\nabla_{\mathbf{u}}\Phi_p(\mathbf{u}_{k+1}) - \nabla_{\mathbf{u}}\Phi(\mathbf{u}_{k+1}, \boldsymbol{\theta}_0)\big)^{\mathsf{T}}, \tag{2.13}$$

$$\boldsymbol{\lambda}_{k+1}^{\mathrm{G}_i} = (\boldsymbol{I}_{n_{\mathrm{u}}} - \boldsymbol{K}^{\mathrm{G}_i})\boldsymbol{\lambda}_k^{\mathrm{G}_i} + \boldsymbol{K}^{\mathrm{G}_i}\big(\nabla_{\mathbf{u}}\mathrm{G}_{p,i}(\mathbf{u}_{k+1}) - \nabla_{\mathbf{u}}\mathrm{G}_i(\mathbf{u}_{k+1}, \boldsymbol{\theta}_0)\big)^{\mathsf{T}}, \quad i = 1, \ldots, n_{\mathrm{g}}, \tag{2.14}$$

where the filter matrices $\boldsymbol{K}^{\varepsilon}$, $\boldsymbol{K}^{\Phi}$ and $\boldsymbol{K}^{\mathrm{G}_i}$ are typically diagonal matrices with eigenvalues in the interval $(0, 1]$; $\boldsymbol{I}$ denotes the identity matrix. Alternatively, one can apply input filters as

$$\mathbf{u}_{k+1} = \mathbf{u}_k + \boldsymbol{K}(\mathbf{u}^{\star}_{k+1} - \mathbf{u}_k), \tag{2.15}$$

where $\boldsymbol{K}$ is the input filter matrix that is a diagonal matrix with eigenvalues in the interval $(0, 1]$; $\mathbf{u}^{\star}_{k+1}$ is the optimal input value obtained at k^{th} RTO iteration by solving (2.10).

2.5.2 Properties of MA

The most appealing feature of MA is its ability to find the plant KKT point upon convergence. This property is summarized by the following theorem.

Theorem 2.1 (In MA, convergence ⇒ plant optimality [83])**.** *Consider the modified model in* (2.10) *with filters on either the modifiers or the inputs as described in* (2.12) *and* (2.15)*. Let* $\mathbf{u}_{\infty} := \lim_{k\to\infty} \mathbf{u}_k$ *be a fixed point reached upon iteratively solving Problem* (2.10) *and also the KKT point of the Problem. Then,* $\mathbf{u}_{\infty}$ *is also a KKT point of the plant problem* (2.1)*.*

Note that MA converges to an input $\mathbf{u}_{\infty}$ that satisfies only the first-order NCOs. However, to guarantee that $\mathbf{u}_{\infty}$ is a local minimum for the plant Problem (2.1), the second-order NCOs must also be verified. This would require estimation of the plant Hessian that remains an open problem, especially in the presence of measurement noise. To this end, nice theoretical conditions are established in [42] that proposes to use second-order modifiers to guarantee meeting the second-order NCOs.

Although the current know-how limits verifying that the plant local minimum $\mathbf{u}^{\star}_p$ is reached in MA, it is always desirable to satisfy model-adequacy conditions. The model-adequacy conditions for MA are described in the following theorem.

Theorem 2.2 (Model-adequacy conditions for MA [83])**.** *Let* $\mathbf{u}^{\star}_p$ *be a regular point for the constraints and the unique plant minimum. Let* $\nabla^2_r \mathcal{L}(\mathbf{u}^{\star}_p)$ *denote the reduced Hessian of the Lagrangian of Problem* (2.10) *at* $\mathbf{u}^{\star}_p$*. Then, the following statements hold*

(a) If $\nabla^2_r \mathcal{L}(\mathbf{u}^{\star}_p)$ *is positive definite, then the process model is adequate for use in the MA scheme.*

(b) If $\nabla^2_r \mathcal{L}(\mathbf{u}^{\star}_p)$ *is not positive semi-definite, then the process model is inadequate for use in the MA scheme.*

(c) If $\nabla^2_r \mathcal{L}(\mathbf{u}^{\star}_p)$ *is positive semi-definite and singular, then the second-order conditions are not conclusive with respect to model adequacy.*

Remark 2.1. *It should be noted that satisfaction of the model-adequacy conditions in* MA *does not guarantee satisfaction of the second-order* NCO*s for the plant upon convergence, as this would require knowledge of the plant Hessian at the converged input* $\mathbf{u}_\infty$*. If the reduced Hessian of the plant Lagrangian is positive semi-definite at* $\mathbf{u}_\infty$*, only then the second-order* NCO*s are established for the plant Problem* (2.1)*. However, satisfaction of model-adequacy conditions by the modified model in* MA *guarantees that, if the plant minimum* $\mathbf{u}_p^\star$ *is reached at an* RTO *iteration by* MA*, then* MA *will converge to* $\mathbf{u}_p^\star$*, that is, not diverge to sub-optimal input values. On the other hand, if model adequacy is not established, then* MA *may well diverge from* $\mathbf{u}_p^\star$ *to sub-optimal points and result in oscillatory behavior.*

2.5.3 The Lagrangian-Modifiers Form

The Lagrange-modifiers form is first proposed in [86]. This form has the advantage that it is parsimonious in the number of modifier elements as the gradient modifier computed for the Lagrangian function is utilized as opposed to the separate gradient modifiers on the cost and constraint functions. This form has been successfully exploited in [105], where using gradient modifiers on the Lagrangian helps in speeding up the convergence of MA:

$$\min_{\mathbf{u}} \quad \Phi_{m,k}(\mathbf{u}) := \Phi(\mathbf{u}, \boldsymbol{\theta}_0) + (\boldsymbol{\lambda}_k^{\mathcal{L}})^\mathsf{T}\mathbf{u} \tag{2.16a}$$

$$\text{s.t.} \quad \mathbf{G}_{m,k}(\mathbf{u}) := \mathbf{G}(\mathbf{u}, \boldsymbol{\theta}_0) + \boldsymbol{\varepsilon}_k^{\mathrm{G}} \leq \mathbf{0}, \tag{2.16b}$$

where $\boldsymbol{\varepsilon}_k^{\mathrm{G}} \in \mathbb{R}^{n_\mathrm{g}}$ are the zeroth-order modifiers for the constraints; $\boldsymbol{\lambda}_k^{\mathcal{L}} \in \mathbb{R}^{n_\mathrm{u}}$ is the first-order modifier; and $\mathbf{G} \in \mathbb{R}^{n_\mathrm{g}}$ is the vector of model constraints. At the k^{th} RTO iteration, the modifiers are computed as follows

$$(\boldsymbol{\lambda}_k^{\mathcal{L}})^\mathsf{T} = \nabla_\mathrm{u}\mathcal{L}_p(\mathbf{u}_k, \boldsymbol{\mu}_k) - \nabla_\mathrm{u}\mathcal{L}(\mathbf{u}_k, \boldsymbol{\mu}_k, \boldsymbol{\theta}_0), \tag{2.17a}$$

$$\boldsymbol{\varepsilon}_k^{\mathrm{G}} = \mathbf{G}_p(\mathbf{u}_k) - \mathbf{G}(\mathbf{u}_k, \boldsymbol{\theta}_0), \tag{2.17b}$$

$$\text{with} \quad \mathcal{L}_p(\mathbf{u}, \boldsymbol{\mu}) = \Phi_p(\mathbf{u}) + \boldsymbol{\mu}^\mathsf{T}\mathbf{G}_p(\mathbf{u}), \tag{2.17c}$$

$$\text{and} \quad \mathcal{L}(\mathbf{u}, \boldsymbol{\mu}, \boldsymbol{\theta}) = \Phi(\mathbf{u}, \boldsymbol{\theta}) + \boldsymbol{\mu}^\mathsf{T}\mathbf{G}(\mathbf{u}, \boldsymbol{\theta}). \tag{2.17d}$$

where $\boldsymbol{\mu} \in \mathbb{R}^{n_\mathrm{g}}$ is the vector of Lagrange multipliers and $(\mathbf{u}_k, \boldsymbol{\mu}_k)$ is the k^{th} realization of the pair $(\mathbf{u}, \boldsymbol{\mu})$.

2.6 Influential and Non-Influential Parameters

The models deployed in RTO are often not well calibrated for the purpose of predicting the plant NCOs. If one decides to make parametric adjustments so as to improve model predictions, then understanding the relationship between the model parameters and the model ability to accurately predict the NCOs is of paramount importance.

To this end, tools for sensitivity analysis play a central role. The goal of sensitivity analysis is to classify the elements of a given set of parameters into two categories. The first category represents the subset of parameters that influences the NCOs and the second category contains the subset that does not. The concept of *influential* parameters is formally defined as follows.

Consider the mapping $y = f(\boldsymbol{\theta})$, $y \in \mathbb{R}$, $\boldsymbol{\theta} \in \mathbb{R}^{n_\theta}$. In addition, consider the spaces $\boldsymbol{\mathcal{I}}$ and $\boldsymbol{\mathcal{NI}}$ that are orthogonal complements to each other such that they form a direct sum on $\mathbb{R}^{n_\theta}$, i.e., $\boldsymbol{\mathcal{NI}} \oplus \boldsymbol{\mathcal{I}} = \mathbb{R}^{n_\theta}$. Then, the influential and non-influential spaces can be defined as:

Definition 2.2 (Influential parameters [123])**.** *On the subspace $\boldsymbol{\mathcal{NI}} \subset \mathbb{R}^{n_\theta}$, a parameter direction $\boldsymbol{\theta}$ is said to be non-influential for a given scalar function $f \in \mathbb{R}$ if $|f(\boldsymbol{\theta}_1) - f(\boldsymbol{\theta}_2)| < \epsilon$ for all $\boldsymbol{\theta}_1, \boldsymbol{\theta}_2 \in \boldsymbol{\mathcal{NI}}$, where ϵ is a positive scalar. The orthogonal complement of $\boldsymbol{\mathcal{NI}} \subset \mathbb{R}^{n_\theta}$ is the subspace $\boldsymbol{\mathcal{I}}$ of influential parameters.*

There exists several tools to find the space of influential parameters. For instance, Fisher information is the classic technique of finding influential parameters. However, the techniques of local sensitivity analysis, such as Fisher information, are limited in their use as they are not representative of the global behavior. The global sensitivity analysis addresses this issue by incorporating sensitivity information from the entire parameter space. Several techniques exist in the literature for global sensitivity analysis [8, 74, 91, 113, 124]. A notable technique of global sensitivity analysis is that of Sobol indices that quantify the global influence of the parameter on the variance of the response [97, 124]. Although this approach has the advantage that it does not require any linearization, the computation of the sensitivity indices can be prohibitively expensive for large parameter dimensions. The alternative is to use linearization techniques to approximate local sensitivities and employ these linearized relations in establishing influential parameters [1, 29]. The approximation of global sensitivities are then obtained by aggregating the local sensitivities evaluated at random parameter values sampled from an admissible parameter set. In the following subsection, one of such approaches, known as active subspaces [28, 29, 110], is recalled.

2.6.1 Active Subspaces

Active subspaces are an emerging set of tools for dimension reduction in the parameter space of a function of several parameters. A low-dimensional active subspace, when present, identifies important directions in the space of parameters. Perturbing the function along the active subspace changes the function more, on average, than perturbing the function orthogonally to the active subspace. This low-dimensional structure provides insights that characterize the dependence of quantities of interest (such as model prediction of NCOs) on parameters $\boldsymbol{\theta}$.

Hence, instead of attempting to identify a collection of parameters, the active subspace method seeks to identify a collection of directions in parameter space. Each direction is a set of weights that define a linear combination of the original elements of the vector $\boldsymbol{\theta}$. If the function does not change as the parameter values move along a particular direction, then, that direction can be safely ignored.

To find the active subspace of a scalar function $f : \mathbb{R}^{n_\theta} \to \mathbb{R}$, the following matrix $\boldsymbol{C} \in \mathbb{R}^{n_\theta \times n_\theta}$ is evaluated

$$\boldsymbol{C} = \int \left(\nabla_{\boldsymbol{\theta}} f(\boldsymbol{\theta})\right)^{\mathsf{T}} \left(\nabla_{\boldsymbol{\theta}} f(\boldsymbol{\theta})\right) \rho \, \mathrm{d}\boldsymbol{\theta}, \tag{2.18}$$

where ρ is the probability density function of $\boldsymbol{\theta}$ over the admissible bounded set Θ with $\rho = 0$ for $\boldsymbol{\theta} \notin \Theta$. Here, the parameter vector $\boldsymbol{\theta}$ is the scaled version of the original parameters; i.e. without loss of generality, the admissible parameter set can be taken as $\Theta = [-1, 1]^{n_\theta} \subseteq \mathbb{R}^{n_\theta}$. Note that $\boldsymbol{C}$ is symmetric and positive semi-definite, so it diagonalizes as

$$\boldsymbol{C} = \boldsymbol{Q} \boldsymbol{\Pi} \boldsymbol{Q}^{\mathsf{T}}, \quad \boldsymbol{\Pi} = \mathrm{diag}(\pi_1, \ldots, \pi_{n_\theta}), \tag{2.19}$$

with $\pi_1 \geq \cdots \geq \pi_{n_\theta} \geq 0$; $\boldsymbol{Q} \in \mathbb{R}^{n_\theta \times n_\theta}$ is an orthonormal matrix whose columns $\mathbf{q}_1, \ldots, \mathbf{q}_{n_\theta}$ are the normalized eigenvectors of $\boldsymbol{C}$. The following lemma highlights the relation between the scalar function f and the eigenvalues of the corresponding matrix $\boldsymbol{C}$.

Lemma 2.3 (Constantine [28]). *The mean-squared directional derivative of f with respect to $\mathbf{q}_l$ satisfies*

$$\int \left(\left(\nabla_{\boldsymbol{\theta}} f(\boldsymbol{\theta})\right) \mathbf{q}_l \right)^2 \rho \, \mathrm{d}\boldsymbol{\theta} = \pi_l, \quad l = 1, \ldots, n_\theta, \tag{2.20}$$

where π_l is the eigenvalue corresponding to the eigenvector $\mathbf{q}_l$ of $\boldsymbol{C}$.

Lemma 2.3 implies that, if an eigenvalue π_l of $\boldsymbol{C}$ is zero, then the corresponding directional derivative $\left(\nabla_{\boldsymbol{\theta}} f(\boldsymbol{\theta})\, \mathbf{q}_l\right)$ is zero everywhere in the domain Θ. Hence, the function f is constant along the direction $\mathbf{q}_l$.

Moreover, if some eigenvalues are relatively *large* and some are *small* such that there exists a clear gap in the spectrum of eigenvalues of $\boldsymbol{C}$, then complementary subspaces of $\mathbb{R}^{n_\theta}$ can be easily defined, where the values of f in one subspace *varies a lot*, while it *varies only a little* along directions contained in the other complementary subspace. More precisely, the matrices $\boldsymbol{\Pi}$ and $\boldsymbol{Q}$ can be written in block form as

$$\boldsymbol{\Pi} = \begin{bmatrix} \boldsymbol{\Pi}_1 & \mathbf{0} \\ \mathbf{0} & \boldsymbol{\Pi}_2 \end{bmatrix}, \text{ and } \quad \boldsymbol{Q} = [\boldsymbol{Q}_1 \quad \boldsymbol{Q}_2], \tag{2.21}$$

where $\boldsymbol{\Pi}_1 \in \mathbb{R}^{m\times m}$ and $\boldsymbol{Q}_1 \in \mathbb{R}^{n_\theta \times m}$ with $m \leq n_\theta$. Provided it exists, the block form is such that $m \ll n_\theta$ and $\pi_m \gg \pi_{m+1}$. Based on this partitioning, we obtain the new (parameter) directions $\boldsymbol{\vartheta}_1 \in \mathbb{R}^m$ and $\boldsymbol{\vartheta}_2 \in \mathbb{R}^{n_\theta - m}$

$$\boldsymbol{\vartheta}_1 := (\boldsymbol{Q}_1)^\mathsf{T}\boldsymbol{\theta}, \text{ and } \boldsymbol{\vartheta}_2 := (\boldsymbol{Q}_2)^\mathsf{T}\boldsymbol{\theta}. \tag{2.22}$$

Motivated from the above, one can write $\boldsymbol{\theta} = \boldsymbol{Q}_1\,\boldsymbol{\vartheta}_1 + \boldsymbol{Q}_2\,\boldsymbol{\vartheta}_2$. In addition, using the chain rule, we have

$$\nabla_{\boldsymbol{\vartheta}_1} f(\boldsymbol{\theta}) = \nabla_{\boldsymbol{\theta}} f(\boldsymbol{\theta})\,\boldsymbol{Q}_1 \quad \text{and} \quad \nabla_{\boldsymbol{\vartheta}_2} f(\boldsymbol{\theta}) = \nabla_{\boldsymbol{\theta}} f(\boldsymbol{\theta})\,\boldsymbol{Q}_2.$$

The next lemma combines the $\boldsymbol{\vartheta}_1 \in \mathbb{R}^m$ and $\boldsymbol{\vartheta}_2 \in \mathbb{R}^{n_\theta - m}$ with Lemma 2.3.

Lemma 2.4 (Constantine [28]). *The mean-squared gradients of f with respect to $\boldsymbol{\vartheta}_1$ and $\boldsymbol{\vartheta}_2$ satisfy*

$$\begin{aligned} \int \big(\nabla_{\boldsymbol{\vartheta}_1} f(\boldsymbol{\theta})\big)\big(\nabla_{\boldsymbol{\vartheta}_1} f(\boldsymbol{\theta})\big)^\mathsf{T} \rho\,\mathrm{d}\boldsymbol{\theta} &= \pi_1 + \cdots + \pi_m, \\ \int \big(\nabla_{\boldsymbol{\vartheta}_2} f(\boldsymbol{\theta})\big)\big(\nabla_{\boldsymbol{\vartheta}_2} f(\boldsymbol{\theta})\big)^\mathsf{T} \rho\,\mathrm{d}\boldsymbol{\theta} &= \pi_{m+1} + \cdots + \pi_{n_\theta}, \end{aligned} \tag{2.23}$$

Assuming that $\pi_{m+1} = \cdots = \pi_{n_\theta} = 0$, the following is obtained:

$$\nabla_{\boldsymbol{\vartheta}_2} f(\boldsymbol{\theta}) = \boldsymbol{0} \quad \forall\,\boldsymbol{\theta} \in \Theta. \tag{2.24}$$

Motivated from the above and provided they exist, the rotated parameters $\boldsymbol{\vartheta}_2$ are termed *the non-influential parameters*. The gradient with respect to $\boldsymbol{\vartheta}_2$ is zero everywhere in the domain. Not surprisingly, the subspace spanned by the columns of $\boldsymbol{Q}_1$ corresponds to *the influential subspace* $\boldsymbol{\mathcal{I}}$, whereas the subspace spanned by the columns of $\boldsymbol{Q}_2$ corresponds to *the non-influential subspace* $\boldsymbol{\mathcal{N}\mathcal{I}}$; i.e.,

$$\boldsymbol{\mathcal{I}} = \mathrm{col}\,(\boldsymbol{Q}_1), \tag{2.25a}$$

$$\boldsymbol{\mathcal{N}\mathcal{I}} = \mathrm{col}\,(\boldsymbol{Q}_2), \tag{2.25b}$$

where $\mathrm{col}\,(\cdot)$ is the column space.[1] Since obtaining $\boldsymbol{C}$ may be analytically prohibitive in certain cases, [28] proposes computing $\boldsymbol{C}$ and the influential spaces via a sampling-based scheme described in Algorithm 2.1.

2.6.2 Computing the Influential Space via Active Subspaces

The process model can be so complex that the analytical expression for the matrix $\boldsymbol{C}$ is intractable. In [28], a sampling-based Monte-Carlo approach is proposed to approx-

[1] Note that [28] refers to (2.25a) as the *active* subspace, and to (2.25b) as the *inactive* subspace.

Algorithm 2.1

Step 1: Draw N independent samples $\boldsymbol{\theta}_j$ from Θ using the probability density ρ.

Step 2: For each sample $\boldsymbol{\theta}_j$, compute the gradient $\nabla_{\boldsymbol{\theta}} f(\boldsymbol{\theta}_j)$.

Step 3: Compute

$$\hat{\boldsymbol{C}} = \frac{1}{N} \sum_{j=1}^{N} \left(\nabla_{\boldsymbol{\theta}} f(\boldsymbol{\theta}_j)\right)^{\mathsf{T}} \left(\nabla_{\boldsymbol{\theta}} f(\boldsymbol{\theta}_j)\right).$$

Step 4: Compute the eigendecomposition of matrix $\hat{\boldsymbol{C}}$ to obtain $\hat{\boldsymbol{Q}}$

$$\hat{\boldsymbol{C}} = \hat{\boldsymbol{Q}}\,\hat{\boldsymbol{\Pi}}\,(\hat{\boldsymbol{Q}})^{\mathsf{T}}, \;\; \hat{\boldsymbol{Q}} = [\hat{\mathbf{q}}_1 \; \cdots \; \hat{\mathbf{q}}_{n_\theta}], \text{ and } \hat{\boldsymbol{\Pi}} = \operatorname{diag}(\hat{\pi}_1, \ldots, \hat{\pi}_{n_\theta}).$$

Step 5: Select the influential and non-influential space by partitioning the matrix $\hat{\boldsymbol{Q}}$

$$\begin{aligned}
&\hat{\boldsymbol{Q}}_1 = [\hat{\mathbf{q}}_1 \; \cdots \; \hat{\mathbf{q}}_m], \quad \hat{\boldsymbol{Q}}_2 = [\hat{\mathbf{q}}_{m+1} \; \cdots \; \hat{\mathbf{q}}_{n_\theta}], \\
&m : \hat{\pi}_m \gg \hat{\pi}_{m+1}\,, \\
&\boldsymbol{\mathcal{I}} = \operatorname{col}(\hat{\boldsymbol{Q}}_1), \quad \boldsymbol{\mathcal{N}\mathcal{I}} = \operatorname{col}(\hat{\boldsymbol{Q}}_2).
\end{aligned}$$

imate $\boldsymbol{C}$. It is recommended to scale the inputs $\mathbf{u}$ and parameters $\boldsymbol{\theta}$ so that they lie between -1 and 1. Based on the approximations $\hat{\boldsymbol{C}}$ and $\hat{\boldsymbol{Q}}$, the influential and non-influential spaces are computed by means of Algorithm 2.1. The algorithm computes the approximation $\hat{\boldsymbol{C}}$ by randomly sampling the parameter space and computing the function gradient at each of the samples. The gradient samples are then aggregated to form the matrix $\hat{\boldsymbol{C}}$ whose eigenvalue decomposition reveals the active subspace.

2.6.3 Sufficient Summary Plots

Sufficient summary plots (SSP), first proposed in [30], are powerful visualization tools that help identify a low-dimensional structure in a function of several parametric quantities. The sufficient summary plot generalizes the idea of plotting the quantity of interest (such as output of a scalar function f) against a linear combination of the parametric variables, $\boldsymbol{\omega}^{\mathsf{T}}\boldsymbol{\theta}$, where $\boldsymbol{\omega} \in \mathbb{R}^{n_\theta}$. If $\boldsymbol{\omega}$ is a canonical basis vector (a vector of zeros with a one in a single entry), then the sufficient summary plot becomes a simple scatter plot, such as those described in Section 1.2.3 in [112]. The choice of the linear combination vector $\boldsymbol{\omega}$ is important, as the right value can reveal a univariate trend, if present. The univariate trend indicates that the quantity of interest can be treated like a function of a single variable given by $\boldsymbol{\omega}^{\mathsf{T}}\boldsymbol{\theta}$.

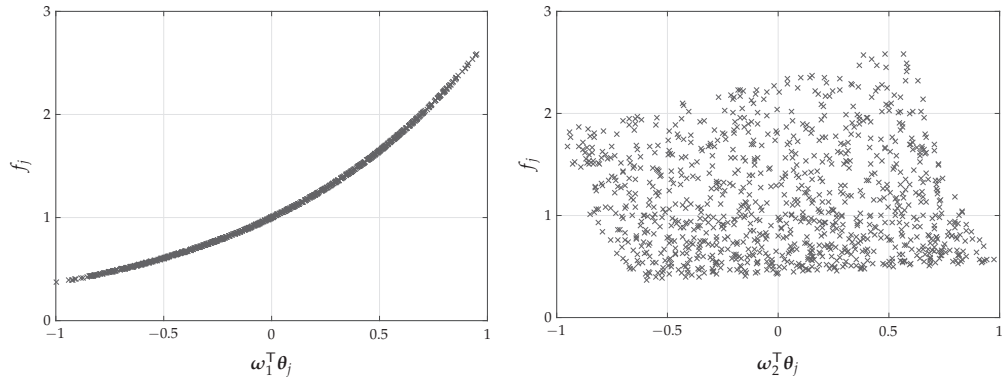


Figure 2.1 – Sufficient summary plots.

Example 2.1. *Consider the following mapping*

$$f = \exp\left(0.8\,\theta_1 + 0.2\,\theta_2\right), \quad \boldsymbol{\theta} = [\theta_1, \theta_2]^\mathsf{T}. \tag{2.26}$$

To illustrate the concept of SSPs, two values for $\boldsymbol{\omega}$ are chosen as $\boldsymbol{\omega}_1 = [0.8, 0.2]^\mathsf{T}$ and $\boldsymbol{\omega}_2 = [-0.2, 0.8]^\mathsf{T}$ with $\boldsymbol{\omega}_1^\mathsf{T}\boldsymbol{\omega}_2 = 0$. Now assume that $\boldsymbol{\theta}$ belongs to the hypercube $[-1, 1]^2$ and is uniformly distributed. The parameter vector $\boldsymbol{\theta}$ is randomly sampled and the function mapping is evaluated at each of the random sample. Each parametric sample $\boldsymbol{\theta}_j$ is then projected on $\boldsymbol{\omega}_1$ and $\boldsymbol{\omega}_2$. The function samples f_j are plotted against the projections in the SSPs plotted in the Figure 2.1. A univariate trend appears in the left plot in Figure 2.1a, whereas no trend emerges when plotted for $\boldsymbol{\omega}_2$ in Figure 2.1b. This implies that the function f can be considered as a function of a single variable.

The parameter directions that can reveal a trend in low dimensions can be found via active subspaces. If the active subspace is computed for the function in Example 2.1, then the space spanned by $\boldsymbol{\omega}_1$ is returned as the influential space $\mathcal{I}$ and the non-influential space $\mathcal{NI}$ is spanned by $\boldsymbol{\omega}_2$. In general, if the active subspace of a function mapping is one- or two-dimensional, then SSPs can be drawn to confirm its presence.

3 Measure only the KKTs that Matter

In the presence of plant-model mismatch, constraint and gradient estimation is pivotal in meeting the plant KKT conditions. However, estimating the plant gradient information requires experiments, where the plant is sequentially perturbed in every input direction. This procedure consumes a lot of time as a continuously operating plant needs to settle to its new steady-state (or a batch operation needs to finish) after each perturbation.

The experimental cost of gradient estimation can be reduced by relying more on the model and understanding how the model gradient behaves when the model parameters are perturbed. This requires parametric sensitivity analysis of the model NCOs, in particular, of the model Lagrangian gradient. Such analysis often yields a partitioning of the input space into two subspaces. One of the subspaces has a high sensitivity to parametric perturbations, while the other has relatively low sensitivity. Therefore, for the purpose of the estimation of the plant gradient, the low-sensitivity input space can be discarded and the plant gradient is estimated only in the highly sensitive input space.

Input-space partitioning can be achieved by local parametric sensitivity analysis, as proposed in [31, 33]. The resulting set of sensitive input directions are called *privileged* directions. When the parameters are locally perturbed, the model-Lagrangian gradient varies a lot along privileged directions and, therefore, relying on the model gradient along these directions causes sub-optimality. Hence, the model gradient is corrected only along privileged directions, that is, the plant gradient is estimated only in the low-dimensional input space spanned by the privileged directions. As a result, the experimental cost of gradient-estimation is significantly reduced. A directional modifier-adaptation (DMA) scheme is proposed in [33] that exploits local parametric sensitivity-based privileged directions. The gradients of the model cost and constraint functions are partially corrected via appropriate modifier terms.

However, when the model dependence on the parameters becomes increasingly

nonlinear, computing privileged directions via local sensitivity analysis may not lead to meeting the plant KKT conditions. Therefore, under large parametric uncertainty, local analysis becomes insufficient in capturing the correct set of privileged directions. This chapter extends the concept of DMA to cover the case where the parametric uncertainty is not local, but belongs to a fairly large uncertainty set. In this case, it is argued that correcting the gradients only in the privileged directions identified offline via local sensitivity analysis may result in significant sub-optimality. Instead, it is proposed here to perform a global sensitivity analysis using ideas derived from active subspaces.

Thus, in addition to measuring the KKT elements corresponding to the local values of the plant constraints that come at no extra experimental cost, this chapter proposes to measure the plant gradient information in the low-dimensional privileged directions that are discovered via global sensitivity analysis. This way, the total number of measured KKT elements are reduced.

This chapter is structured as follows. Sections 3.1 and 3.2 present the background explaining the concept of privileged directions followed by a recap of the DMA algorithm. Section 3.3 presents the mathematical tools required to compute the privileged directions via global sensitivity analysis. Section 3.4 proposes an RTO methodology—active directional modifier-adaptation (ADMA) algorithm that discovers privileged directions computed via global sensitivity analysis. Section 3.5 compares different privileged-direction-based modifier-adaptation schemes in simulation studies of two different batch-to-batch optimization problems. Section 3.6 discusses the computational complexity of performing global sensitivity analysis and proposes two alternative methods for discovering privileged directions that are computationally cheaper. Section 3.6 also compares one of the alternative methods to the previously proposed method of computing privileged directions on the same simulation studies described in Section 3.5. Finally, Section 3.7 summarizes the chapter.

3.1 Concept of Privileged Directions

The main idea behind privileged directions is to find the set of a few input directions along which the model Lagrangian gradient is highly sensitive to parametric variations. Therefore, instead of correcting the full gradient, only the directional derivative is corrected along a few input-directions. The dimension n_r of the privileged directions space is usually small compared to the input dimension n_u. Hence, estimating the plant directional derivative require fewer finite-difference-steps leading to a reduction in the experimental cost of plant gradient estimation. To this end, directional derivative is defined as follows

Definition 3.1 (Directional Derivative)**.** *Consider a continuously differentiable function $f(\mathbf{u})$. The directional derivative of f in any direction $\mathbf{r} \in \mathbb{R}^{n_\mathrm{r}}$ contained in an input sub-*

space is defined as

$$\nabla_W f(\mathbf{u}) := \left.\frac{\partial f(\mathbf{u}+\boldsymbol{W}\mathbf{r})}{\partial \mathbf{r}}\right|_{\mathbf{r}=\mathbf{0}}, \tag{3.1}$$

with $\nabla_W f \in \mathbb{R}^{1\times n_\mathrm{r}}$*, and* $\boldsymbol{W} \in \mathbb{R}^{n_\mathrm{u}\times n_\mathrm{r}}$ *is a matrix with rank* n_r*. Note that*

$$\nabla_W f(\mathbf{u}) = \nabla_\mathbf{u} f(\mathbf{u})\boldsymbol{W}. \tag{3.2}$$

Example 3.1. *To illustrate the concept of privileged directions, consider the following unconstrained optimization problem*

$$\mathcal{L}(\mathbf{u},\boldsymbol{\theta}) = (2\,\theta_1+\theta_2)\,u_1u_2+\theta_3 \quad \textit{with } \mathbf{u}=[u_1,\,u_2]^\mathsf{T} \textit{ and } \boldsymbol{\theta}=[\theta_1,\,\theta_2,\,\theta_3]^\mathsf{T}. \tag{3.3}$$

The Lagrangian gradient at $\mathbf{u}_1=[1,\,1]^\mathsf{T}$ *reads*

$$\nabla_\mathbf{u}\mathcal{L}(\mathbf{u}_1) = [2\,\theta_1+\theta_2,\,2\,\theta_1+\theta_2].$$

The directional derivatives in the input directions $[-1,\,1]^\mathsf{T}$ *and* $[1,\,1]^\mathsf{T}$ *are*

$$\nabla_\mathbf{u}\mathcal{L}(\mathbf{u}_1)\,[-1,\,1]^\mathsf{T} = 0 \quad \textit{and} \quad \nabla_\mathbf{u}\mathcal{L}(\mathbf{u}_1)\,[1,\,1]^\mathsf{T} = 2\,(2\,\theta_1+\theta_2).$$

Evidently, at $\mathbf{u}_1$*, the gradient is highly sensitive to parametric perturbations along the input direction* $[1,\,1]^\mathsf{T}$*, whereas it is insensitive along the input direction* $[-1,\,1]^\mathsf{T}$*. Thus, correcting the model gradient uniquely along the privileged direction* $[1,\,1]^\mathsf{T}$ *is sufficient.*

3.2 Directional Modifier Adaptation

In Directional Modifier Adaptation (DMA)[1], the sensitivity of the model Lagrangian gradient is evaluated with respect to local parametric variations. This local parametric sensitivity is evaluated only once at the model optimum and thus, the procedure is performed offline. In DMA, the local sensitivity matrix $\boldsymbol{A}^\star \in \mathbb{R}^{n_\mathrm{u}\times n_\theta}$ is computed as follows:

$$\boldsymbol{A}^\star := \nabla_{\mathbf{u}\boldsymbol{\theta}}\mathcal{L}(\mathbf{u}^\star,\boldsymbol{\mu}^\star,\boldsymbol{\theta}_0) = \left.\frac{\partial^2\mathcal{L}}{\partial\boldsymbol{\theta}\,\partial\mathbf{u}}\right|_{\mathbf{u}^\star,\boldsymbol{\mu}^\star,\boldsymbol{\theta}_0}, \tag{3.4}$$

where $\boldsymbol{\mu}^\star$ are the Lagrange multipliers corresponding to the nominal solution $\mathbf{u}^\star$. Singular value decomposition of $\boldsymbol{A}^\star$ gives

$$\boldsymbol{A}^\star = \boldsymbol{W}\boldsymbol{S}\boldsymbol{V}^\mathsf{T}, \tag{3.5}$$

[1]DMA is **not** a contribution of this thesis. It was originally proposed by Costello et al. in [33].

Algorithm 3.1 Directional Modifier Adaptation (DMA) [33].

Step 0 (Initialization): Compute the nominal solution $\mathbf{u}^\star$ and the corresponding Lagrange multipliers $\boldsymbol{\mu}^\star$ by solving Problem (2.2) for $\boldsymbol{\theta} = \boldsymbol{\theta}_0$. Evaluate the sensitivity matrix $\boldsymbol{A}^\star$ in (3.4), perform singular value decomposition and determine the privileged directions $\boldsymbol{W}_\mathrm{r}$.
Set the initial values of the modifiers to zero, $\boldsymbol{\varepsilon}_0^\mathrm{G} = \mathbf{0}$, $\boldsymbol{\lambda}_0^\Phi = \mathbf{0}$ and $\boldsymbol{\lambda}_0^{\mathrm{G}_i} = \mathbf{0}$, and the values of the filter matrices $\boldsymbol{K}^\varepsilon$, $\boldsymbol{K}^\Phi$ and $\boldsymbol{K}^{\mathrm{G}_i}$ (typically diagonal matrices) with eigenvalues in the interval $(0, 1]$. Also, set arbitrarily $\mathbf{u}_0 = \mathbf{0}$.

for $k = 0 \to \infty$.

Step 1 (Optimization): Solve the modified model-based Problem (2.10) for $\boldsymbol{\theta} = \boldsymbol{\theta}_0$ to compute the optimal inputs $\mathbf{u}_{k+1}$.

Step 2 (Plant evaluation): Apply $\mathbf{u}_{k+1}$ to the plant and collect the measurements $\mathbf{y}_p(\mathbf{u}_{k+1})$. Use these measurements to compute $\Phi_p(\mathbf{u}_{k+1})$ and $\mathbf{G}_p(\mathbf{u}_{k+1})$.

Step 3 (Estimation of directional derivatives): Estimate the directional derivative of the plant cost $\nabla_{W_\mathrm{r}}\Phi_p(\mathbf{u}_{k+1})$ and of the constraints $\nabla_{W_\mathrm{r}}\mathrm{G}_{p,i}(\mathbf{u}_{k+1})$, $i = 1, \ldots, n_\mathrm{g}$, as per (3.4)-(3.1). At $\mathbf{u}_{k+1}$, the full gradients are computed as

$$\widehat{\nabla_\mathbf{u}\Xi_p}(\mathbf{u}_{k+1}) := \nabla_\mathbf{u}\Xi(\mathbf{u}_{k+1}, \boldsymbol{\theta}_0)\left(\boldsymbol{I}_{n_\mathrm{u}} - \boldsymbol{W}_\mathrm{r}\boldsymbol{W}_\mathrm{r}^+\right) + \nabla_{W_\mathrm{r}}\Xi_p(\mathbf{u}_{k+1})\boldsymbol{W}_\mathrm{r}^+,$$

with $\Xi \in \{\Phi, \mathrm{G}_i\}$, and $\boldsymbol{W}_\mathrm{r}^+$ the Moore-Penrose pseudo-inverse of $\boldsymbol{W}_\mathrm{r}$.

Step 4 (Modifier update): Update the modifiers using first-order filters

$$\begin{aligned}
\boldsymbol{\varepsilon}_{k+1}^\mathrm{G} &= (\boldsymbol{I}_{n_\mathrm{g}} - \boldsymbol{K}^\varepsilon)\boldsymbol{\varepsilon}_k^\mathrm{G} + \boldsymbol{K}^\varepsilon\left(\mathbf{G}_p(\mathbf{u}_{k+1}) - \mathbf{G}(\mathbf{u}_{k+1}, \boldsymbol{\theta}_0)\right),\\
\boldsymbol{\lambda}_{k+1}^\Phi &= (\boldsymbol{I}_{n_\mathrm{u}} - \boldsymbol{K}^\Phi)\boldsymbol{\lambda}_k^\Phi + \boldsymbol{K}^\Phi\left(\widehat{\nabla_\mathbf{u}\Phi_p}(\mathbf{u}_{k+1}) - \nabla_\mathbf{u}\Phi(\mathbf{u}_{k+1}, \boldsymbol{\theta}_0)\right)^\mathsf{T},\\
\boldsymbol{\lambda}_{k+1}^{\mathrm{G}_i} &= (\boldsymbol{I}_{n_\mathrm{u}} - \boldsymbol{K}^{\mathrm{G}_i})\boldsymbol{\lambda}_k^{\mathrm{G}_i} + \boldsymbol{K}^{\mathrm{G}_i}\left(\widehat{\nabla_\mathbf{u}\mathrm{G}_{p,i}}(\mathbf{u}_{k+1}) - \nabla_\mathbf{u}\mathrm{G}_i(\mathbf{u}_{k+1}, \boldsymbol{\theta}_0)\right)^\mathsf{T}, \quad i = 1, \ldots, n_\mathrm{g}.
\end{aligned}$$

end

where $\boldsymbol{W} \in \mathbb{R}^{n_\mathrm{u} \times n_\mathrm{u}}$ is an orthonormal matrix whose columns $\mathbf{w}_i$, $i = 1, \ldots, n_\mathrm{u}$, are the left singular vectors of $\boldsymbol{A}^\star$; $\boldsymbol{S} \in \mathbb{R}^{n_\mathrm{u} \times n_\theta}$ is a rectangular diagonal matrix whose diagonal elements s_i, $i = 1, \ldots, n_\mathrm{s}$ with $n_\mathrm{s} = \min\{n_\mathrm{u}, n_\theta\}$, are the singular values of $\boldsymbol{A}^\star$; and $\boldsymbol{V} \in \mathbb{R}^{n_\theta \times n_\theta}$. Through the singular values of $\boldsymbol{A}^\star$, one can rank the input directions $\mathbf{w}_i$ according to their sensitivity with respect to local parametric perturbations. The reduced matrix $\boldsymbol{W}_\mathrm{r} \in \mathbb{R}^{n_\mathrm{u} \times n_\mathrm{r}}$, with $n_\mathrm{r} < n_\mathrm{u}$, can be constructed as

$$\boldsymbol{W}_\mathrm{r} = [\mathbf{w}_1 \cdots \mathbf{w}_{n_\mathrm{r}}] : \quad \mathrm{s}_{n_\mathrm{r}+1} \ll \mathrm{s}_{n_\mathrm{r}}, \tag{3.6}$$

i.e., a large gap between the consecutive singular values is exploited to construct $\boldsymbol{W}_\mathrm{r}$. At each RTO iteration, the directional derivatives are estimated only in the *privileged* di-

rections spanned by the columns of $\boldsymbol{W}_{\mathrm{r}}$. Note that the number of privileged directions for DMA satisfies the following condition

$$n_{\mathrm{r}} \leq \min\{n_{\mathrm{u}}, n_{\theta}\}. \tag{3.7}$$

The DMA scheme is summarized in Algorithm 3.1. Past studies have shown a significant reduction in the experimental cost of gradient estimation when DMA is applied. For instance, DMA has been applied to perform RTO on an airbone-wind energy system [32, 33]. Therein, DMA significantly reduces the input space from 40 to 2 dimensions for the purpose of gradient estimation. Yet, the optimality loss is only 5 percent despite adapting the gradients in only two directions (in the other 38 directions, nominal model gradients are used).

Nevertheless, by no means can it be expected that a local sensitivity analysis would systematically yield a good approximation to global sensitivities. Moreover, DMA computes the privileged directions only once at the model optimum. But, during the RTO iterations, the value of input $\mathbf{u}$ and the Lagrange multipliers $\boldsymbol{\mu}$ change, and thereby, the model Lagrangian sensitivity changes. Hence, the privileged directions found at the model optimum is no more the correct set of directions. In such a case, adapting the gradients in the privileged directions found offline by DMA may result in significant optimality loss. In order to address this issue, we propose an *online* procedure for determining the privileged directions via a *global* sensitivity analysis carried out at each RTO iteration.

3.3 Discovering Privileged Directions via Global Sensitivity Analysis

To discover privileged directions under large parametric uncertainty, sensitivity concepts similar to the ones that form the backbone of active subspaces are utilized [118, 119]. In active subspaces the parameters are perturbed to discover a *low-dimensional structure in the parameter space* that is responsible for the most variability in the model output. Here too the parameters are perturbed. However, instead of exploring parameter space, *a low-dimensional structure in the input space* is discovered. This low-dimensional input subspace is such that along it, the projection of the Lagrangian gradient of the model varies a lot. Therefore, the model gradient can not be trusted in this low-dimensional subspace and hence, plant measurements are required to make appropriate corrections.

Consider a twice differentiable function $f : \mathcal{U} \times \Theta \to \mathbb{R}$, where $\mathcal{U} \subseteq \mathbb{R}^{n_{\mathrm{u}}}$, $\Theta \subseteq \mathbb{R}^{n_{\theta}}$ and Θ is a bounded and connected set. Let the probability density function of $\boldsymbol{\theta}$ be $\rho(\boldsymbol{\theta})$. Also, consider that $\rho(\boldsymbol{\theta})$ is strictly positive and bounded for $\boldsymbol{\theta} \in \Theta$ and

$\rho(\boldsymbol{\theta}) = 0$ for $\boldsymbol{\theta} \notin \Theta$, so that the focus is only on the parameter values of interest. Assume that ρ and Θ are such that the components of $\boldsymbol{\theta}$ are independent with mean zero and scaled according to their range. Such a normalization ensures that each parameter component is given equal importance. In addition, assume that the matrix $\nabla_{\mathbf{u}\boldsymbol{\theta}} f(\mathbf{u}, \boldsymbol{\theta}) := \frac{\partial^2 f(\mathbf{u},\boldsymbol{\theta})}{\partial \boldsymbol{\theta}\, \partial \mathbf{u}} \in \mathbb{R}^{n_\mathrm{u} \times n_\theta}$ is bounded, that is,

$$||\nabla_{\mathbf{u}\boldsymbol{\theta}} f(\mathbf{u}, \boldsymbol{\theta})|| \leq L, \quad L > 0 \qquad \forall\, \mathbf{u} \in \mathcal{U},\ \boldsymbol{\theta} \in \Theta,$$

where $||\cdot||$ is the Frobenius norm.

Next, consider the matrix $\boldsymbol{A}_k \in \mathbb{R}^{n_\mathrm{u} \times n_\mathrm{u}}$ as

$$\boldsymbol{A}_k = \int_\Theta \left(\nabla_{\mathbf{u}\boldsymbol{\theta}} f(\mathbf{u}_k, \boldsymbol{\theta})\right)\left(\nabla_{\mathbf{u}\boldsymbol{\theta}} f(\mathbf{u}_k, \boldsymbol{\theta})\right)^\mathsf{T} \rho \, \mathrm{d}\boldsymbol{\theta}. \tag{3.8}$$

It follows that each element of $\boldsymbol{A}_k$ is the average of the product of partial double derivatives (it is assumed that the partial double derivatives exist)

$$\mathrm{a}_{ij,k} = \int_\Theta \sum_{l=1}^{n_\theta} \left(\frac{\partial^2 f(\mathbf{u}, \boldsymbol{\theta})}{\partial \theta_l\, \partial u_i}\bigg|_{\mathbf{u}_k}\right)\left(\frac{\partial^2 f(\mathbf{u}, \boldsymbol{\theta})}{\partial \theta_l\, \partial u_j}\bigg|_{\mathbf{u}_k}\right) \rho \, \mathrm{d}\boldsymbol{\theta}, \quad i, j = 1, \ldots, n_\mathrm{u}, \tag{3.9}$$

where $\mathrm{a}_{ij,k}$ is the (i, j) element of $\boldsymbol{A}_k$; θ_l is the l^{th} element of $\boldsymbol{\theta}$; and u_i is the i^{th} element of $\mathbf{u}$. The matrix $\boldsymbol{A}_k$ is positive semi-definite since

$$\boldsymbol{\nu}^\mathsf{T} \boldsymbol{A}_k\, \boldsymbol{\nu} = \int_\Theta \left(\boldsymbol{\nu}^\mathsf{T}\, \nabla_{\mathbf{u}\boldsymbol{\theta}} f(\mathbf{u}_k, \boldsymbol{\theta})\right)\left(\boldsymbol{\nu}^\mathsf{T}\, \nabla_{\mathbf{u}\boldsymbol{\theta}} f(\mathbf{u}_k, \boldsymbol{\theta})\right)^\mathsf{T} \rho \, \mathrm{d}\boldsymbol{\theta} \geq 0 \quad \forall\, \boldsymbol{\nu} \in \mathbb{R}^{n_\mathrm{u}}.$$

Moreover, as $\boldsymbol{A}_k$ is symmetric, we can write

$$\boldsymbol{A}_k = \boldsymbol{W}_k\, \boldsymbol{\Sigma}_k\, \boldsymbol{W}_k^\mathsf{T}, \quad \boldsymbol{\Sigma}_k = \mathrm{diag}(\sigma_{1,k}, \ldots, \sigma_{n_\mathrm{u},k}), \quad \sigma_{1,k} \geq \cdots \geq \sigma_{n_\mathrm{u},k} \geq 0, \tag{3.10}$$

where $\boldsymbol{W}_k \in \mathbb{R}^{n_\mathrm{u} \times n_\mathrm{u}}$ is an orthonormal matrix whose columns $\mathbf{w}_{i,k}$, $i = 1, \ldots, n_\mathrm{u}$, are the normalized eigenvectors of $\boldsymbol{A}_k$.

Lemma 3.1 (Singhal et al. [119])**.** *For all $\mathbf{u}_k \in \mathcal{U}$, it holds that*

$$\int_\Theta \left|\left|\mathbf{w}_{i,k}^\mathsf{T}\, \nabla_{\mathbf{u}\boldsymbol{\theta}} f(\mathbf{u}_k, \boldsymbol{\theta})\right|\right|^2 \rho \, \mathrm{d}\boldsymbol{\theta} = \sigma_{i,k}, \quad i = 1, \ldots, n_\mathrm{u}, \tag{3.11}$$

where $\sigma_{i,k}$ is the eigenvalue corresponding to the eigenvector $\mathbf{w}_{i,k}$ of $\boldsymbol{A}_k$.

Proof. The definition of $\sigma_{i,k}$ implies

$$\sigma_{i,k} = \mathbf{w}_{i,k}^\mathsf{T}\, \boldsymbol{A}_k\, \mathbf{w}_{i,k},$$

which can be written as

$$\begin{aligned}\sigma_{i,k} &= \mathbf{w}_{i,k}^{\mathsf{T}}\left(\int_{\Theta}\left(\nabla_{\mathbf{u}\boldsymbol{\theta}} f(\mathbf{u}_k,\boldsymbol{\theta})\right)\left(\nabla_{\mathbf{u}\boldsymbol{\theta}} f(\mathbf{u}_k,\boldsymbol{\theta})\right)^{\mathsf{T}}\rho\,\mathrm{d}\boldsymbol{\theta}\right)\mathbf{w}_{i,k}\\ &= \int_{\Theta}\left(\mathbf{w}_{i,k}^{\mathsf{T}}\nabla_{\mathbf{u}\boldsymbol{\theta}} f(\mathbf{u}_k,\boldsymbol{\theta})\right)\left(\mathbf{w}_{i,k}^{\mathsf{T}}\nabla_{\mathbf{u}\boldsymbol{\theta}} f(\mathbf{u}_k,\boldsymbol{\theta})\right)^{\mathsf{T}}\rho\,\mathrm{d}\boldsymbol{\theta}\\ &= \int_{\Theta}\left|\left|\mathbf{w}_{i,k}^{\mathsf{T}}\nabla_{\mathbf{u}\boldsymbol{\theta}} f(\mathbf{u}_k,\boldsymbol{\theta})\right|\right|^2\rho\,\mathrm{d}\boldsymbol{\theta}.\end{aligned}$$

□

It follows from this lemma that, if the eigenvalue $\sigma_{i,k} = 0$, then

$$\mathbf{w}_{i,k}^{\mathsf{T}}\nabla_{\mathbf{u}\boldsymbol{\theta}} f(\mathbf{u}_k,\boldsymbol{\theta}) = \mathbf{0}, \quad \forall\,\boldsymbol{\theta}\in\Theta. \tag{3.12}$$

Integrating (3.12) with respect to $\boldsymbol{\theta}$, and using the fundamental theorem of calculus, gives

$$\sigma_{i,k} = 0, \implies \nabla_{\mathbf{u}} f(\mathbf{u}_k,\boldsymbol{\theta})\,\mathbf{w}_{i,k} = \mathrm{c}, \quad \mathrm{c}\in\mathbb{R}, \quad \forall\,\boldsymbol{\theta}\in\Theta. \tag{3.13}$$

In other words, the lemma implies that the directional derivative of f (with respect to $\mathbf{u}$ at $\mathbf{u}_k$) in the direction $\mathbf{w}_{i,k}$ is constant regardless of the value of the parameter $\boldsymbol{\theta}$ (as long as $\boldsymbol{\theta}\in\Theta$).

The matrix $\boldsymbol{W}_k$ can be split into two submatrices, the matrix $\boldsymbol{W}_{1,k}\in\mathbb{R}^{n_{\mathrm{u}}\times n_{\mathrm{r}}}$ and the matrix $\boldsymbol{W}_{2,k}\in\mathbb{R}^{n_{\mathrm{u}}\times(n_{\mathrm{u}}-n_{\mathrm{r}})}$, whereby $\boldsymbol{W}_{1,k}$ contains the eigenvectors $\mathbf{w}_{i,k}$ corresponding to the n_{r} non-zero eigenvalues and the matrix $\boldsymbol{W}_{2,k}$ collects the remaining eigenvectors corresponding to the zero eigenvalues.

$$\begin{aligned}&\boldsymbol{W}_k = \left[\boldsymbol{W}_{1,k}\ \boldsymbol{W}_{2,k}\right],\\ &\boldsymbol{W}_{1,k} = \left[\mathbf{w}_{1,k}\ \cdots\ \mathbf{w}_{n_{\mathrm{r}},k}\right] : \sigma_{1,k}\geq\cdots\geq\sigma_{n_{\mathrm{r}},k}>0, \quad n_{\mathrm{r}}\leq n_{\mathrm{u}},\\ &\boldsymbol{W}_{2,k} = \left[\mathbf{w}_{n_{\mathrm{r}}+1,k}\ \cdots\ \mathbf{w}_{n_{\mathrm{u}},k}\right] : \sigma_{n_{\mathrm{r}}+1,k}=\cdots=\sigma_{n_{\mathrm{u}},k}=0.\end{aligned} \tag{3.14}$$

The directional derivative $\nabla_{\boldsymbol{W}_{1,k}} f$ computed at $\mathbf{u}_k$ can not be trusted as it is highly sensitive to the parametric perturbations. On the other hand, parametric perturbations have minimal impact on the directional derivative $\nabla_{\boldsymbol{W}_{2,k}} f$ computed at $\mathbf{u}_k$. Therefore, input directions given by the columns of the matrix $\boldsymbol{W}_{1,k}$ are chosen as the set of privileged directions.

Comparison of Active Subspaces and Privileged Directions

At this point, privileged directions computed via global sensitivity analysis may be confused with the active subspaces. Indeed, the derivation of active subspaces in Lemma 2.3 is similar to Lemma 3.1. Both the privileged directions and the active

Table 3.1 – Comparison of Active Subspaces and Privileged Directions.

Active subspaces	**Privileged directions via global sensitivity analysis**
The goal is to find only few directions in $\mathbb{R}^{n_\theta}$ that best describes variability in f upon perturbing $\boldsymbol{\theta}$	The goal is to find only few directions in $\mathbb{R}^{n_\mathrm{u}}$ that best describes variability in $\nabla_\mathrm{u} f$ upon perturbing $\boldsymbol{\theta}$
$\boldsymbol{\theta}$ is considered a random variable	$\boldsymbol{\theta}$ is considered a random variable
Sensitivity $\nabla_\mathrm{u} f$ is not considered	Sensitivity $\nabla_\mathrm{u} f$ local in $\mathbf{u}$ is considered
Low-dimensional structure in $\boldsymbol{\theta}$ is discovered	Low-dimensional structure in $\mathbf{u}$ is discovered

subspaces involve finding sensitivity with respect to the random variable $\boldsymbol{\theta}$.

However, active subspace concentrate exclusively on finding a low-dimensional structure in the *parameter space* that best quantify the variability of the output of a given function f to large parametric perturbations. Moreover, in the computation of active subspace the sensitivity $\nabla_\mathrm{u} f$ with respect to the input variable is not considered.

On the other hand, privileged directions found via global sensitivity analysis additionally deals with the local sensitivity in the input $\mathbf{u}$. Here, input variable $\mathbf{u}$ is not considered a random variable and is independent of $\boldsymbol{\theta}$. A low-dimensional structure is discovered in the *input space* that locally quantifies the variability of the function output to large parametric perturbations. The main features of the two concepts are summarized in Table 3.1.

3.4 Active Directional Modifier Adaptation

Ideally, privileged directions should be chosen such that they capture the maximum variability of the Lagrangian gradient with respect to parametric perturbations [118, 119]. As parametric perturbations get large, the local sensitivity analysis conducted in DMA may not be able to yield such directions. Therefore, it is proposed to find the set of privileged directions based on the following global sensitivity matrix $\boldsymbol{A}_k$

$$\boldsymbol{A}_k = \int_\Theta \left(\nabla_{\mathbf{u}\boldsymbol{\theta}} \mathcal{L}(\mathbf{u}_k, \boldsymbol{\mu}_k, \boldsymbol{\theta})\right)\left(\nabla_{\mathbf{u}\boldsymbol{\theta}} \mathcal{L}(\mathbf{u}_k, \boldsymbol{\mu}_k, \boldsymbol{\theta})\right)^\mathsf{T} \rho \, \mathrm{d}\boldsymbol{\theta}. \tag{3.15}$$

Then, the resulting privileged direction matrix $\boldsymbol{W}_{1,k}$ in (3.14) is used to update the modifiers as follows

$$\boldsymbol{\varepsilon}_k^{\mathrm{G}} = \mathbf{G}_p(\mathbf{u}_k) - \mathbf{G}(\mathbf{u}_k, \boldsymbol{\theta}_0), \tag{3.16a}$$

$$(\boldsymbol{\lambda}_k^{\Phi})^{\mathsf{T}} = \widehat{\nabla_{\mathbf{u}}\Phi_p}(\mathbf{u}_k) - \nabla_{\mathbf{u}}\Phi(\mathbf{u}_k, \boldsymbol{\theta}_0), \tag{3.16b}$$

$$(\boldsymbol{\lambda}_k^{\mathrm{G}})^{\mathsf{T}} = \widehat{\nabla_{\mathbf{u}}\mathbf{G}_p}(\mathbf{u}_k) - \nabla_{\mathbf{u}}\mathbf{G}(\mathbf{u}_k, \boldsymbol{\theta}_0), \tag{3.16c}$$

where the gradients $\widehat{\nabla_{\mathbf{u}}\Phi_p}(\mathbf{u}_k)$ and $\widehat{\nabla_{\mathbf{u}}\mathbf{G}_p}(\mathbf{u}_k)$ are updated as

$$\widehat{\nabla_{\mathbf{u}}\Xi_p}(\mathbf{u}_k) = \nabla_{\mathbf{u}}\Xi(\mathbf{u}_k, \boldsymbol{\theta}_0)(\boldsymbol{I}_{n_{\mathrm{u}}} - \boldsymbol{W}_{1,k}\boldsymbol{W}_{1,k}^{+}) + \nabla_{\boldsymbol{W}_{1,k}}\Xi_p(\mathbf{u}_k)\boldsymbol{W}_{1,k}^{+}, \quad \Xi \in \{\Phi, \mathrm{G}_i\}, \tag{3.17}$$

where $\nabla_{\boldsymbol{W}_{1,k}}\Xi_p(\mathbf{u}_k)$ is the directional derivative of Ξ_p at $\mathbf{u}_k$ in the directions given by the columns of the matrix $\boldsymbol{W}_{1,k}$, and $\boldsymbol{W}_{1,k}^{+}$ is the Moore-Penrose pseudo-inverse of $\boldsymbol{W}_{1,k}$. For the sake of simplicity, the filter matrices used in Algorithm 3.1 are dropped from the modifiers in (3.16). Updating the modifiers using first-order filters does not affect the validity of the results presented hereafter.

3.4.1 KKT Matching under Parametric Plant-Model Mismatch

Further developments are based on the following technical assumptions.

Assumption 3.1 (Parametric plant-model mismatch)**.** *Let $\boldsymbol{\theta}_p \in \Theta$ be the vector of true plant parameters such that*

$$\Phi(\mathbf{u}, \boldsymbol{\theta}_p) = \Phi_p(\mathbf{u}), \tag{3.18a}$$

$$\mathbf{G}(\mathbf{u}, \boldsymbol{\theta}_p) = \mathbf{G}_p(\mathbf{u}), \tag{3.18b}$$

where Θ is a bounded and connected set in which $\boldsymbol{\theta}$ lies with the probability density function ρ.

Assumption 3.2 (Exact plant directional derivatives)**.** *At each RTO iteration k, exact plant directional derivatives are available for the cost and constraint functions in the directions given by the columns of the matrix $\boldsymbol{W}_{1,k}$.*

Assumption 3.3 (Exact sensitivity information)**.** *The matrix $\boldsymbol{A}_k$ in* (3.15) *is exactly known at each RTO iteration k.*

Theorem 3.2. (Plant optimality upon convergence) *Consider the optimization Problem* (2.10) *with the modifiers* (3.16) *and the gradient updates* (3.17) *with $\boldsymbol{W}_{1,k}$ satisfying* (3.14)*. Let Assumptions 3.1-3.3 hold. Also, assume that $\boldsymbol{\theta}_0 \in \Theta$. If the iterative solution to this problem converges to the fixed point $(\mathbf{u}_\infty, \boldsymbol{\varepsilon}_\infty^{\mathrm{G}}, \boldsymbol{\lambda}_\infty^{\Phi}, \boldsymbol{\lambda}_\infty^{\mathrm{G}})$, with $\mathbf{u}_\infty$ being a KKT point of Problem* (2.10)*, then $\mathbf{u}_\infty$ is also a KKT point for the plant Problem* (2.1)*.*

Proof. The modifiers take the following values upon convergence to $\mathbf{u}_\infty$

$$\boldsymbol{\varepsilon}^{\mathrm{G}}_\infty = \mathbf{G}_p(\mathbf{u}_\infty) - \mathbf{G}(\mathbf{u}_\infty, \boldsymbol{\theta}_0), \tag{3.19a}$$
$$(\boldsymbol{\lambda}^{\Phi}_\infty)^{\mathsf{T}} = \widehat{\nabla_{\mathbf{u}}\Phi_p}(\mathbf{u}_\infty) - \nabla_{\mathbf{u}}\Phi(\mathbf{u}_\infty, \boldsymbol{\theta}_0), \tag{3.19b}$$
$$(\boldsymbol{\lambda}^{\mathrm{G}}_\infty)^{\mathsf{T}} = \widehat{\nabla_{\mathbf{u}}\mathbf{G}_p}(\mathbf{u}_\infty) - \nabla_{\mathbf{u}}\mathbf{G}(\mathbf{u}_\infty, \boldsymbol{\theta}_0). \tag{3.19c}$$

The KKT conditions at $\mathbf{u}_\infty$ for Problem (2.10) read

$$\mathbf{G}_{m,\infty}(\mathbf{u}_\infty) \leq \mathbf{0}, \tag{3.20a}$$
$$\boldsymbol{\mu}^{\mathsf{T}}_\infty \mathbf{G}_{m,\infty}(\mathbf{u}_\infty) = 0, \quad \boldsymbol{\mu}_\infty \geq \mathbf{0}, \tag{3.20b}$$
$$\nabla_{\mathbf{u}}\Phi_{m,\infty}(\mathbf{u}_\infty) + \boldsymbol{\mu}_\infty \nabla_{\mathbf{u}}\mathbf{G}_{m,\infty}(\mathbf{u}_\infty) = \mathbf{0}. \tag{3.20c}$$

From (2.10), (3.19) and (3.20), one can write

$$\mathbf{G}_p(\mathbf{u}_\infty) \leq \mathbf{0}, \tag{3.21a}$$
$$\boldsymbol{\mu}^{\mathsf{T}}_\infty \mathbf{G}_p(\mathbf{u}_\infty) = 0, \quad \boldsymbol{\mu}_\infty \geq \mathbf{0}, \tag{3.21b}$$
$$\widehat{\nabla_{\mathbf{u}}\Phi_p}(\mathbf{u}_\infty) + \boldsymbol{\mu}^{\mathsf{T}}_\infty \widehat{\nabla_{\mathbf{u}}\mathbf{G}_p}(\mathbf{u}_\infty) = \mathbf{0}. \tag{3.21c}$$

Next, consider (3.17) at $\mathbf{u}_\infty$

$$\widehat{\nabla_{\mathbf{u}}\Phi_p}(\mathbf{u}_\infty) = \nabla_{\mathbf{u}}\Phi(\mathbf{u}_\infty, \boldsymbol{\theta}_0)(\boldsymbol{I}_{n_{\mathrm{u}}} - \boldsymbol{W}_{1,\infty}\boldsymbol{W}^{+}_{1,\infty}) + \nabla_{\boldsymbol{W}_{1,\infty}}\Phi_p(\mathbf{u}_\infty)\boldsymbol{W}^{+}_{1,\infty}.$$

It follows from (3.2) that

$$\begin{aligned}
\widehat{\nabla_{\mathbf{u}}\Phi_p}(\mathbf{u}_\infty) &= \nabla_{\mathbf{u}}\Phi(\mathbf{u}_\infty, \boldsymbol{\theta}_0)(\boldsymbol{I}_{n_{\mathrm{u}}} - \boldsymbol{W}_{1,\infty}\boldsymbol{W}^{+}_{1,\infty}) + \nabla_{\mathbf{u}}\Phi_p(\mathbf{u}_\infty)\boldsymbol{W}_{1,\infty}\boldsymbol{W}^{+}_{1,\infty}, \\
\widehat{\nabla_{\mathbf{u}}\Phi_p}(\mathbf{u}_\infty) &= \nabla_{\mathbf{u}}\Phi(\mathbf{u}_\infty, \boldsymbol{\theta}_0)(\boldsymbol{W}_\infty\boldsymbol{W}^{\mathsf{T}}_\infty - \boldsymbol{W}_{1,\infty}\boldsymbol{W}^{+}_{1,\infty}) + \nabla_{\mathbf{u}}\Phi_p(\mathbf{u}_\infty)\boldsymbol{W}_{1,\infty}\boldsymbol{W}^{+}_{1,\infty}, \\
\widehat{\nabla_{\mathbf{u}}\Phi_p}(\mathbf{u}_\infty) &= \nabla_{\mathbf{u}}\Phi(\mathbf{u}_\infty, \boldsymbol{\theta}_0)(\boldsymbol{W}_{1,\infty}\boldsymbol{W}^{\mathsf{T}}_{1,\infty} + \boldsymbol{W}_{2,\infty}\boldsymbol{W}^{\mathsf{T}}_{2,\infty} - \boldsymbol{W}_{1,\infty}\boldsymbol{W}^{+}_{1,\infty}) \\
&\quad + \nabla_{\mathbf{u}}\Phi_p(\mathbf{u}_\infty)\boldsymbol{W}_{1,\infty}\boldsymbol{W}^{+}_{1,\infty}.
\end{aligned}$$

Since the matrix $\boldsymbol{W}_{1,\infty}$ has orthonormal columns, $\boldsymbol{W}^{+}_{1,\infty} = \boldsymbol{W}^{\mathsf{T}}_{1,\infty}$, and

$$\widehat{\nabla_{\mathbf{u}}\Phi_p}(\mathbf{u}_\infty) = \nabla_{\mathbf{u}}\Phi(\mathbf{u}_\infty, \boldsymbol{\theta}_0)\boldsymbol{W}_{2,\infty}\boldsymbol{W}^{\mathsf{T}}_{2,\infty} + \nabla_{\mathbf{u}}\Phi_p(\mathbf{u}_\infty)\boldsymbol{W}_{1,\infty}\boldsymbol{W}^{\mathsf{T}}_{1,\infty}. \tag{3.22}$$

Similarly, for the constraints, one can write

$$\widehat{\nabla_{\mathbf{u}}\mathbf{G}_p}(\mathbf{u}_\infty) = \nabla_{\mathbf{u}}\mathbf{G}(\mathbf{u}_\infty, \boldsymbol{\theta}_0)\boldsymbol{W}_{2,\infty}\boldsymbol{W}^{\mathsf{T}}_{2,\infty} + \nabla_{\mathbf{u}}\mathbf{G}_p(\mathbf{u}_\infty)\boldsymbol{W}_{1,\infty}\boldsymbol{W}^{\mathsf{T}}_{1,\infty}. \tag{3.23}$$

Using (3.22), (3.23) and (3.21c) gives:

$$\nabla_{\mathbf{u}}\mathcal{L}(\mathbf{u}_\infty, \boldsymbol{\mu}_\infty, \boldsymbol{\theta}_0)\boldsymbol{W}_{2,\infty}\boldsymbol{W}^{\mathsf{T}}_{2,\infty} + \nabla_{\mathbf{u}}\mathcal{L}_p(\mathbf{u}_\infty, \boldsymbol{\mu}_\infty)\boldsymbol{W}_{1,\infty}\boldsymbol{W}^{\mathsf{T}}_{1,\infty} = \mathbf{0}, \tag{3.24}$$

where $\mathcal{L}_p(\mathbf{u}_\infty, \boldsymbol{\mu}_\infty) = \Phi_p(\mathbf{u}_\infty) + \boldsymbol{\mu}^{\mathsf{T}}_\infty \mathbf{G}_p(\mathbf{u}_\infty)$.

We know from Lemma 3.1 that the directional derivatives of the Lagrangian are constant in the directions given by the columns of the matrix $\boldsymbol{W}_{2,\infty}$, since the corresponding eigenvalues are zero. We also know that these directional derivatives are independent of the parameter values $\boldsymbol{\theta} \in \Theta$. Hence,

$$\nabla_{\mathbf{u}}\mathcal{L}(\mathbf{u}_\infty, \boldsymbol{\mu}_\infty, \boldsymbol{\theta}_0)\boldsymbol{W}_{2,\infty} = \nabla_{\mathbf{u}}\mathcal{L}(\mathbf{u}_\infty, \boldsymbol{\mu}_\infty, \boldsymbol{\theta}_p)\boldsymbol{W}_{2,\infty} = \mathbf{c} \in \mathbb{R}^{1\times(n_\mathrm{u}-n_\mathrm{r})}. \tag{3.25}$$

It follows from (3.24), (3.25) and Assumption 3.1 that

$$\nabla_{\mathbf{u}}\mathcal{L}_p(\mathbf{u}_\infty, \boldsymbol{\mu}_\infty)(\boldsymbol{W}_{2,\infty}\boldsymbol{W}_{2,\infty}^\mathsf{T} + \boldsymbol{W}_{1,\infty}\boldsymbol{W}_{1,\infty}^\mathsf{T}) = \mathbf{0}, \tag{3.26a}$$

$$\nabla_{\mathbf{u}}\mathcal{L}_p(\mathbf{u}_\infty, \boldsymbol{\mu}_\infty)(\boldsymbol{W}_\infty\boldsymbol{W}_\infty^\mathsf{T}) = \mathbf{0}, \tag{3.26b}$$

$$\nabla_{\mathbf{u}}\mathcal{L}_p(\mathbf{u}_\infty, \boldsymbol{\mu}_\infty) = \mathbf{0}. \tag{3.26c}$$

Then, from Equations (3.21a), (3.21b) and (3.26c), we conclude that $\mathbf{u}_\infty$ is a KKT point of the plant Problem (2.1). □

Remark 3.1. *An implicit assumption in Theorem 3.2 is that the matrix $\boldsymbol{A}_k$ has at least one eigenvalue equal to zero. However, it may happen in practice that none of the eigenvalues is exactly zero. Instead, some eigenvalues are small compared to others and can be discarded without much information loss. Note that, if the matrix $\boldsymbol{A}_k$ has no eigenvalues that can be discarded, the partitioning* (3.14) *gives $\boldsymbol{W}_{1,k} = \boldsymbol{W}_k$, and* ADMA *reduces to standard* MA.

3.4.2 Practical Aspects of ADMA

This section discusses some of the features of the ADMA algorithm that are pivotal to its performance in practical applications.

Computation of $\hat{\boldsymbol{A}}_k$ and $\hat{\boldsymbol{W}}_k$

The available process models are often too complex for allowing the derivation of analytical expressions for the matrices $\boldsymbol{A}_k$ and $\boldsymbol{W}_k$. For these cases, we propose to estimate these matrices from data using a sampling-based Monte-Carlo approach. It is recommended to scale the inputs and parameters so that they lie between -1 and 1. The estimates $\hat{\boldsymbol{A}}_k$ and $\hat{\boldsymbol{W}}_k$ are computed via the steps given in Algorithm 3.2. The sample size N should be chosen such that increasing N has a negligible effect on the eigenvalues of the matrix $\hat{\boldsymbol{A}}_k$.

Formal Difference Between DMA and ADMA

An alternative approach based on singular value decomposition (SVD) can be used to compute the eigenpairs of $\hat{\boldsymbol{A}}_k$. Note that we can write $\hat{\boldsymbol{A}}_k = \hat{\boldsymbol{B}}_k\hat{\boldsymbol{B}}_k^\mathsf{T}$, where the matrix

Algorithm 3.2 Computation of matrices $\hat{\boldsymbol{A}}_k$ and $\hat{\boldsymbol{W}}_k$

Step 1: Draw N independent samples $\boldsymbol{\theta}_j$ from Θ using the probability density ρ.

Step 2: Compute the $(n_{\mathrm{u}} \times n_\theta)$-dimensional sensitivity matrix of the Lagrangian gradient (for example, via forward finite differences)

$$\nabla_{\mathbf{u}\boldsymbol{\theta}} \mathcal{L}_k^{(j)} := \frac{\partial^2 \mathcal{L}}{\partial \boldsymbol{\theta}\, \partial \mathbf{u}}(\mathbf{u}_k, \boldsymbol{\mu}_k, \boldsymbol{\theta}_j), \quad j = 1, \ldots, N.$$

Step 3: Compute $\hat{\boldsymbol{A}}_k$ as follows

$$\hat{\boldsymbol{A}}_k = \frac{1}{N} \sum_{j=1}^{N} \left(\nabla_{\mathbf{u}\boldsymbol{\theta}} \mathcal{L}_k^{(j)}\right)\left(\nabla_{\mathbf{u}\boldsymbol{\theta}} \mathcal{L}_k^{(j)}\right)^{\mathsf{T}}. \tag{3.27}$$

Step 4: Compute the eigenvalue decomposition of $\hat{\boldsymbol{A}}_k$ to obtain $\hat{\boldsymbol{W}}_k$

$$\hat{\boldsymbol{A}}_k = \hat{\boldsymbol{W}}_k \hat{\boldsymbol{\Sigma}}_k \hat{\boldsymbol{W}}_k^{\mathsf{T}}.$$

$\hat{\boldsymbol{B}}_k \in \mathbb{R}^{n_{\mathrm{u}} \times n_\theta N}$ is

$$\hat{\boldsymbol{B}}_k = \frac{1}{\sqrt{N}} \left[\nabla_{\mathbf{u}\boldsymbol{\theta}} \mathcal{L}_k^{(1)} \cdots \nabla_{\mathbf{u}\boldsymbol{\theta}} \mathcal{L}_k^{(N)}\right]. \tag{3.28}$$

Applying SVD to $\hat{\boldsymbol{B}}_k$ and using the well-known relation of SVD to eigenvalue decomposition gives:

$$\hat{\boldsymbol{B}}_k = \hat{\boldsymbol{W}}_k \hat{\boldsymbol{S}}_k \hat{\boldsymbol{V}}_k^{\mathsf{T}}, \quad \text{with } \hat{\boldsymbol{S}}_k \hat{\boldsymbol{S}}_k^{\mathsf{T}} = \hat{\boldsymbol{\Sigma}}_k. \tag{3.29}$$

This allows comparing the SVDs performed in DMA and in ADMA. In the former, SVD is performed on the sensitivity matrix $\boldsymbol{A}^\star$ that is evaluated for the nominal parameters $\boldsymbol{\theta}_0$. In the latter, SVD is performed on the matrix $\hat{\boldsymbol{B}}_k$ that stacks the local sensitivity matrices evaluated at N randomly chosen realizations of the parametric uncertainty into a single matrix, thereby representing global sensitivity.

The sensitivity matrix $\boldsymbol{A}^\star$ is local in both the inputs $\mathbf{u}$ and the parameters $\boldsymbol{\theta}$. In contrast, the sensitivity matrix $\hat{\boldsymbol{A}}_k$ (or $\hat{\boldsymbol{B}}_k$) is local in the inputs $\mathbf{u}$ but global in the parameters $\boldsymbol{\theta}$. If the model Lagrangian is linear in the parameters, then the sensitivity matrices $\boldsymbol{A}^\star$ and $\hat{\boldsymbol{A}}_k$ are equal when computed for the same inputs, that is, when computed at $(\mathbf{u}^\star, \boldsymbol{\mu}^\star) = (\mathbf{u}_k, \boldsymbol{\mu}_k)$.

Example 3.2. *To verify whether $\hat{\boldsymbol{A}}_k$ successfully captures the global sensitivities with respect to $\boldsymbol{\theta}$, in particular when the model Lagrangian is a nonlinear function of the parameters, and to compare the performance of $\hat{\boldsymbol{A}}_k$ to that of $\boldsymbol{A}^\star$, let us consider the following exemplary La-*

grangian function

$$\mathcal{L}(\mathbf{u}, \boldsymbol{\theta}) = \exp(\theta_1 u_1 + \theta_2 u_2) + \theta_3^2 (u_3 + u_4) + \theta_2 (0.5u_3 - u_4), \tag{3.30a}$$

$$-1 \leq u_i \leq 1, \quad i = 1, \ldots, 4. \tag{3.30b}$$

The only constraints are the input bounds that are obviously independent of the parameters $\boldsymbol{\theta}$. Therefore, the dependency of the Lagrangian on the Lagrange multipliers $\boldsymbol{\mu}$ is omitted in this example.

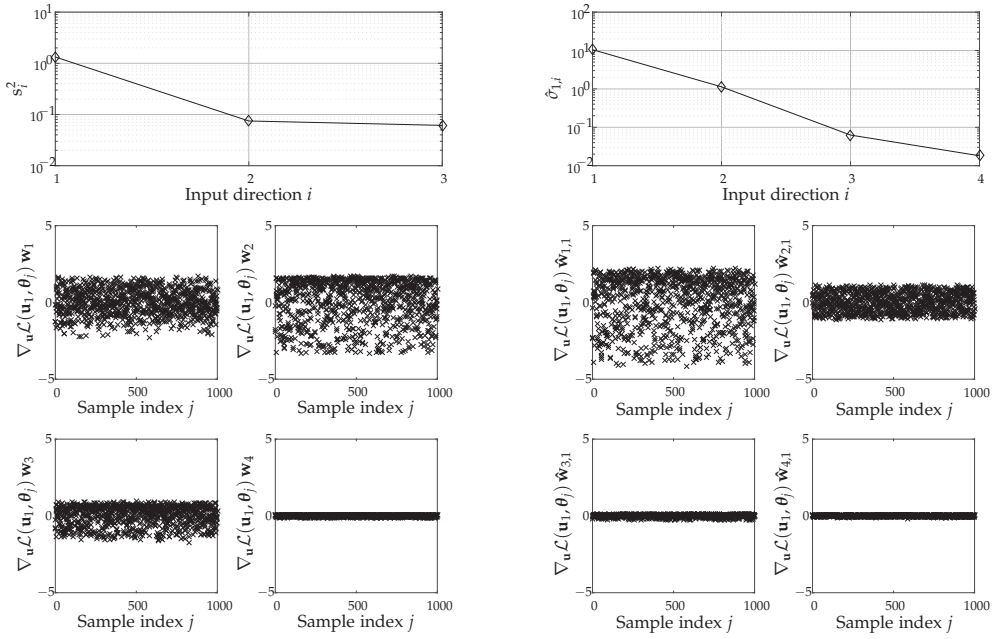
It is assumed that all the elements of the vector $\boldsymbol{\theta} = [\theta_1,\ \theta_2,\ \theta_3]^\mathsf{T}$ are uniformly distributed in the interval $[-2, 0]$. The sensitivity matrix $\boldsymbol{A}^\star$ is constructed from the knowledge of $\boldsymbol{\theta}_0 = [-0.5,\ -0.5,\ -0.1]^\mathsf{T}$ and $\mathbf{u}^\star = \mathbf{u}_1 = [1,\ 1,\ 1,\ -1]^\mathsf{T}$. The matrix $\hat{\boldsymbol{A}}_1$ is constructed using Algorithm 3.2 on the basis of $N = 1000$ Monte-Carlo samples. Note that, as $n_\theta = 3$ and $n_\mathrm{u} = 4$, therefore, the singular value decomposition performed on $\boldsymbol{A}^\star \in \mathbb{R}^{4\times 3}$ give three singular values and four left singular vectors, whereas the eigenvalue decomposition of $\hat{\boldsymbol{A}}_1 \in \mathbb{R}^{4\times 4}$ obviously results in 4 eigenvalues and 4 eigenvectors.

Ideally, the magnitude of the singular value (eigenvalue) should quantify the sensitivity of the directional derivative computed in the direction given by the singular vector (eigenvector). That is, if the parameters are perturbed, then the directional derivative along the singular vector (eigenvector) with largest singular value (eigenvalue) should have the largest variance.[2]

To test if the singular values actually order the singular vectors according to their variance, the gradient $\nabla_\mathbf{u}\mathcal{L}(\mathbf{u}_1, \boldsymbol{\theta}_j)$ is evaluated for all 1000 Monte-Carlo samples. Each sampled gradient is then projected onto each of the left singular vectors of $\boldsymbol{A}^\star$. The resulting projections are plotted in the middle and bottom plots of Figure 3.1a. These plots represent the sensitivities of the directional derivatives to parametric perturbations, with the vertical width of each plot being a measure of variance. The squared singular values are plotted in the top plot of Figure 3.1a. As seen in Figure 3.1a, the directional derivative along the singular vector $\mathbf{w}_1$ does not have the highest variance. In fact, the directional derivative along $\mathbf{w}_2$ has the highest variance. Thus, the two singular vectors are not correctly ordered as the magnitude of the singular values do not correspond to the variance magnitudes along the vectors.

The sampled gradients are also projected onto the eigenvectors of $\hat{\boldsymbol{A}}_1$ (the left singular vectors of $\hat{\boldsymbol{B}}_1$) and the resulting projections are shown in the middle and the bottom plots of Figure 3.1b. The eigenvalues of $\hat{\boldsymbol{A}}_1$ are plotted in the top plot of Figure 3.1b. Here, the eigenvalue magnitude quantifies the parametric sensitivity of $\nabla_\mathbf{u}\mathcal{L}$ in the direction given by the corresponding eigenvector. One sees that the eigenvectors of $\hat{\boldsymbol{A}}_1$ are ranked correctly by the corresponding eigenvalues.

[2]Note that, as the parameters are considered to be random variables, the model-Lagrangian gradient $\nabla_\mathrm{u}\mathcal{L}$ is a random variable. Hence, the directional derivatives are random variables as well.



(a) (b)

Figure 3.1 – (a) **Top plot:** Squared singular values of matrix $A^\star$. **Middle and bottom plots:** Directional derivatives computed at θ_j, $j = 1,\ldots,N$, along the left singular vectors of $A^\star$. The plotted data is mean centered. (b) **Top plot:** Eigenvalues of the matrix $\hat{A}_1$. **Middle and bottom plots:** Directional derivatives computed at θ_j, $j = 1,\ldots,N$, along the eigenvectors of $\hat{A}_1$. The plotted data is mean centered.

Choice of Privileged Directions

The aforementioned analysis indicates that the privileged directions can also be chosen based on the variance of the directional derivatives. The variance along the direction $\mathbf{d} \in \mathbb{R}^{n_\mathrm{u}}$ is computed as

$$\mathrm{Var}\left(\nabla_{\mathbf{u}}\mathcal{L}_k\,\mathbf{d}\right) = \frac{1}{N}\sum_{j=1}^{N}\left|\nabla_{\mathbf{u}}\mathcal{L}_k^{(j)}\,\mathbf{d} - \mathrm{m}\right|^2 \quad \text{with} \quad \mathrm{m} = \frac{1}{N}\sum_{j=1}^{N}\nabla_{\mathbf{u}}\mathcal{L}_k^{(j)}\,\mathbf{d}. \tag{3.31}$$

For the numerical example at hand, the variance is computed from $N = 1000$ Monte-Carlo samples and plotted in Figure 3.2. One sees that the variance does not decrease monotonically for the left singular vectors of $A^\star$, whereas it decreases monotonically for the eigenvectors of $\hat{A}_1$. A monotonic variance decrease indicates that the eigenvalues are ranking the eigenvectors in the right order. To determine the privileged directions, a threshold value on the variance is fixed at 10^{-2}, and the directions that result in a variance larger than the threshold value are chosen as privileged directions. A variance smaller than 10^{-2} indicates that the parametric changes do not cause a

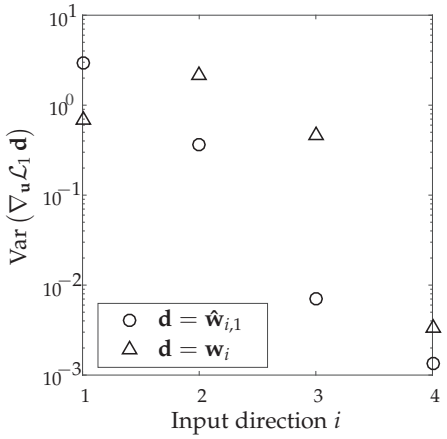


Figure 3.2 – Variance plot for the sensitivity matrices $A^\star$ and $\hat{A}_1$ computed for the example (3.30).

significant change in the gradients along that direction and, therefore, the gradient errors along that direction are relatively small. The global sensitivity matrix $\hat{A}_1$ yields $n_\mathrm{r} = 2$ privileged directions, namely $\hat{\mathbf{w}}_{1,1}$ and $\hat{\mathbf{w}}_{2,1}$, whereas the local sensitivity matrix $A^\star$ yields $n_\mathrm{r} = 3$ privileged directions, namely $\mathbf{w}_1$, $\mathbf{w}_2$ and $\mathbf{w}_3$. Hence, the global sensitivity matrix $\hat{A}_1$ finds a smaller set of privileged directions, thereby reducing the number of required plant experiments for gradient estimation. At the same time, $\hat{A}_1$ ensures that the gradient errors due to parametric perturbations are small along the neglected directions.

Often, one selects the maximal number of privileged directions, n_max, so as to upperbound the experimental budget per RTO iteration. Then, on the basis of the eigenvalues $\hat{\sigma}_{i,k}$, the variance $\mathrm{Var}\left(\nabla_{\mathbf{u}} \mathcal{L}_k \, \mathbf{d}\right)$ and n_max, one can choose the number of privileged directions n_r using one of the following two criteria:

Criterion 1

$$n_\mathrm{r} = \min\{i, n_\mathrm{max}\} : \hat{\sigma}_{i+1,k} \ll \hat{\sigma}_{i,k} \,, \tag{3.32}$$

Criterion 2

$$n_\mathrm{r} = \min\{i, n_\mathrm{max}\} : \mathrm{Var}\left(\nabla_{\mathbf{u}} \mathcal{L}_k \, \hat{\mathbf{w}}_{i,k}\right) \geq \mathrm{v}_\mathrm{min} \text{ and } \mathrm{Var}\left(\nabla_{\mathbf{u}} \mathcal{L}_k \, \hat{\mathbf{w}}_{i+1,k}\right) < \mathrm{v}_\mathrm{min} \,, \tag{3.33}$$

where n_max and the threshold variance v_min are user-defined parameters. The matrix of privileged directions $\hat{\boldsymbol{W}}_{1,k}$ then becomes

$$\hat{\boldsymbol{W}}_{1,k} = [\hat{\mathbf{w}}_{1,k} \cdots \hat{\mathbf{w}}_{n_\mathrm{r},k}]. \tag{3.34}$$

Evaluation of Plant Directional Derivatives

In practice, one often relies on finite differences to evaluate the plant gradients. To reduce the number of plant experiments for gradient estimation, well-excited plant measurements obtained at past RTO iterations can be used. To this end, the optimization objective and the gradient estimation objective are combined by enforcing duality constraints at the RTO layer [15, 84]. Furthermore, estimating gradients from noisy measurements can be reliably achieved by quadratic-approximation of the plant mapping [58]. For a comparative study of different gradient estimation techniques in RTO, we recommend the paper by [81]. The different approaches mentioned here can also be exploited to estimate the plant directional derivatives defined in (3.1).

Structural Plant-Model Mismatch

Assumption 3.1 regarding parametric plant-model mismatch may not be met in practice. In the case of structural plant-model mismatch, ADMA still drives the plant toward optimality in the subspace given by the privileged directions. The following theorem does not require Assumption 3.1.

Theorem 3.3. (Plant optimality limited to privileged directions) *Consider the optimization Problem* (2.10) *with the modifiers* (3.16) *and the gradient updates* (3.17). *Also, assume that* $\boldsymbol{\theta}_0 \in \Theta$. *Let Assumptions 3.2 and 3.3 hold. If the iterative solution to this problem converges to the fixed point* $\left(\mathbf{u}_\infty, \boldsymbol{\varepsilon}^{\mathrm{G}}_\infty, \boldsymbol{\lambda}^{\Phi}_\infty, \boldsymbol{\lambda}^{\mathrm{G}}_\infty\right)$, *with* $\mathbf{u}_\infty$ *being the KKT point of Problem* (2.10), *then* $\mathbf{u}_\infty$ *is also a KKT point for the plant in the directions given by the columns of the matrix* $\boldsymbol{W}_{1,\infty}$.

Proof. See [33]. □

The Algorithm

The proposed ADMA algorithm is summarized in Algorithm 3.3.

3.5 Simulation Studies

The proposed approach is illustrated next via the simulation of two semi-batch reactors.

3.5.1 Williams-Otto Semi-Batch Reactor

ADMA is applied to the problem of run-to run (batch-to-batch) optimization of the Williams-Otto semi-batch reactor described in [111, 132]. The following reactions take

Algorithm 3.3 Active Directional Modifier Adaptation (ADMA)

Step 0 (Initialization): Set the initial values of the modifiers to zero, $\boldsymbol{\varepsilon}_0^{\mathrm{G}} = \mathbf{0}$, $\boldsymbol{\lambda}_0^{\Phi} = \mathbf{0}$ and $\boldsymbol{\lambda}_0^{\mathrm{G}_i} = \mathbf{0}$, and the values of the filter matrices $\boldsymbol{K}^{\varepsilon}$, $\boldsymbol{K}^{\Phi}$ and $\boldsymbol{K}^{\mathrm{G}_i}$ (typically diagonal matrices) with eigenvalues in the interval $(0, 1]$. Also, set arbitrarily $\mathbf{u}_0 = \mathbf{0}$ and select values for $n_{\max}$ and $\mathrm{v}_{\min}$. and set the values of $n_{\max}$ and $\mathrm{v}_{\min}$. Scale the parameters $\boldsymbol{\theta}$ such that the scaled parametric uncertainty range is $[-\mathbf{1}, \mathbf{1}]$.

for $k = 0 \to \infty$

Step 1 (Optimization): Solve the modified optimization Problem (2.10) for $\boldsymbol{\theta} = \boldsymbol{\theta}_0$ to generate the optimal inputs $\mathbf{u}_{k+1}$ and the corresponding Lagrange multipliers $\boldsymbol{\mu}_{k+1}$.

Step 2 (Plant evaluation): Apply $\mathbf{u}_{k+1}$ to the plant and collect the measurements $\mathbf{y}_p(\mathbf{u}_{k+1})$. Use these measurements to evaluate $\Phi_p(\mathbf{u}_{k+1})$ and $\mathbf{G}_p(\mathbf{u}_{k+1})$.

Step 3 (Computation of privileged directions): Compute $\hat{\boldsymbol{W}}_{k+1}$ using Algorithm 3.2 and the privileged direction matrix $\hat{\boldsymbol{W}}_{1,k+1}$ using either Criterion 1 in (3.32) or Criterion 2 in (3.33).

Step 4 (Estimation of directional derivatives): Estimate the directional derivatives of the plant cost $\nabla_{\hat{W}_{1,k+1}} \Phi_p(\mathbf{u}_{k+1})$ and of the constraints $\nabla_{\hat{W}_{1,k+1}} \mathrm{G}_{p,i}(\mathbf{u}_{k+1})$, $i = 1, \ldots, n_{\mathrm{g}}$. At $\mathbf{u}_{k+1}$, the full gradients are computed as

$$\widehat{\nabla_{\mathbf{u}} \Xi_p}(\mathbf{u}_{k+1}) := \nabla_{\mathbf{u}} \Xi(\mathbf{u}_{k+1}, \boldsymbol{\theta}_0)(\boldsymbol{I}_{n_{\mathrm{u}}} - \hat{\boldsymbol{W}}_{1,k+1} \hat{\boldsymbol{W}}_{1,k+1}^{+}) + \nabla_{\hat{W}_{1,k+1}} \Xi_p(\mathbf{u}_{k+1}) \hat{\boldsymbol{W}}_{1,k+1}^{+},$$

with $\Xi \in \{\Phi, \mathrm{G}_i\}$, and $\hat{\boldsymbol{W}}_{1,k+1}^{+}$ the Moore-Penrose pseudo-inverse of $\hat{\boldsymbol{W}}_{1,k+1}$.

Step 5 (Modifier update): Update the modifiers using first-order filters

$$\begin{aligned}
\boldsymbol{\varepsilon}_{k+1}^{\mathrm{G}} &= (\boldsymbol{I}_{n_{\mathrm{g}}} - \boldsymbol{K}^{\varepsilon}) \boldsymbol{\varepsilon}_k^{\mathrm{G}} + \boldsymbol{K}^{\varepsilon} \big(\mathbf{G}_p(\mathbf{u}_{k+1}) - \mathbf{G}(\mathbf{u}_{k+1}, \boldsymbol{\theta}_0) \big), \\
\boldsymbol{\lambda}_{k+1}^{\Phi} &= (\boldsymbol{I}_{n_{\mathrm{u}}} - \boldsymbol{K}^{\Phi}) \boldsymbol{\lambda}_k^{\Phi} + \boldsymbol{K}^{\Phi} \big(\widehat{\nabla_{\mathbf{u}} \Phi_p}(\mathbf{u}_{k+1}) - \nabla_{\mathbf{u}} \Phi(\mathbf{u}_{k+1}, \boldsymbol{\theta}_0) \big)^{\mathsf{T}}, \\
\boldsymbol{\lambda}_{k+1}^{\mathrm{G}_i} &= (\boldsymbol{I}_{n_{\mathrm{u}}} - \boldsymbol{K}^{\mathrm{G}_i}) \boldsymbol{\lambda}_k^{\mathrm{G}_i} + \boldsymbol{K}^{\mathrm{G}_i} \big(\widehat{\nabla_{\mathbf{u}} \mathrm{G}_{p,i}}(\mathbf{u}_{k+1}) - \nabla_{\mathbf{u}} \mathrm{G}_i(\mathbf{u}_{k+1}, \boldsymbol{\theta}_0) \big)^{\mathsf{T}}, \quad i = 1, \ldots, n_{\mathrm{g}}.
\end{aligned}$$

end

place in the reactor

$$\begin{aligned}
A + B &\xrightarrow{k_1} C, \\
C + B &\xrightarrow{k_2} P + E, \\
P + C &\xrightarrow{k_3} G.
\end{aligned}$$

Reactant A is initially present in the reactor, whereas reactant B is continuously fed during the batch. As a result of the exothermic reactions, the desired products P and E and the side product G are formed. The heat generated by the exothermic reactions is removed via a cooling jacket, whose temperature is controlled by manipulating the cooling water temperature. The model equations and parameter values are given in the Appendix A.1.

The objective is to maximize the revenue generated by selling the products produced at the end of the batch, while respecting path constraints on the inlet flowrate of reactant B (F_B), the reactor temperature (T_r), the reactor volume (V) and the cooling water temperature (T_w). The manipulated variables are the time-varying profiles $F_B(t)$ and $T_w(t)$. The dynamic optimization problem can be written mathematically as follows

$$\max_{F_B(t),T_w(t)} \; P_P\, m_P(t_\mathrm{f}) + P_E\, m_E(t_\mathrm{f}) \tag{3.35a}$$

$$\text{s.t.} \quad \text{model equations (A.1)} \tag{3.35b}$$

$$0 \leq F_B(t) \leq 5.784\,\mathrm{kg/s} \tag{3.35c}$$

$$V(t) \leq 5\,\mathrm{m}^3 \tag{3.35d}$$

$$20\,^\circ\mathrm{C} \leq T_w(t) \leq 100\,^\circ\mathrm{C} \tag{3.35e}$$

$$60\,^\circ\mathrm{C} \leq T_r(t) \leq 90\,^\circ\mathrm{C}. \tag{3.35f}$$

The batch time t_f is fixed at 1000 seconds. Problem (3.35) is transformed into a Nonlinear Program (NLP) via direct single shooting. This is done by discretizing the input in time over n_{cs} control stages. For each time interval, the dynamic input variables are parametrized using low-order polynomials. Each time-varying input is parametrtized using $n_{cs} = 40$ piecewise-constant values. This results in the input dimension $n_\mathrm{u} = 80$.

In this simulation study, the plant is substituted by a simulated reality. The simulated reality is then treated as a black box and it is assumed that the concentration measurements of the formed products are available only at the final batch time t_f. This permits the simulated reality to act as a real system/plant.

The experimental cost of evaluating the plant gradients via finite differences is found as follows

Experimental cost per RTO iteration = Total no. of privileged directions $\times$ Batch time.

This implies that the cost to evaluate the full plant gradients is $n_\mathrm{u} \times t_\mathrm{f}$. However, with the DMA and ADMA algorithms, only the derivatives in the privileged directions need to be evaluated, with the experimental cost $n_\mathrm{r} \times t_\mathrm{f}$.

Plant-model mismatch is introduced by considering parametric uncertainty in the

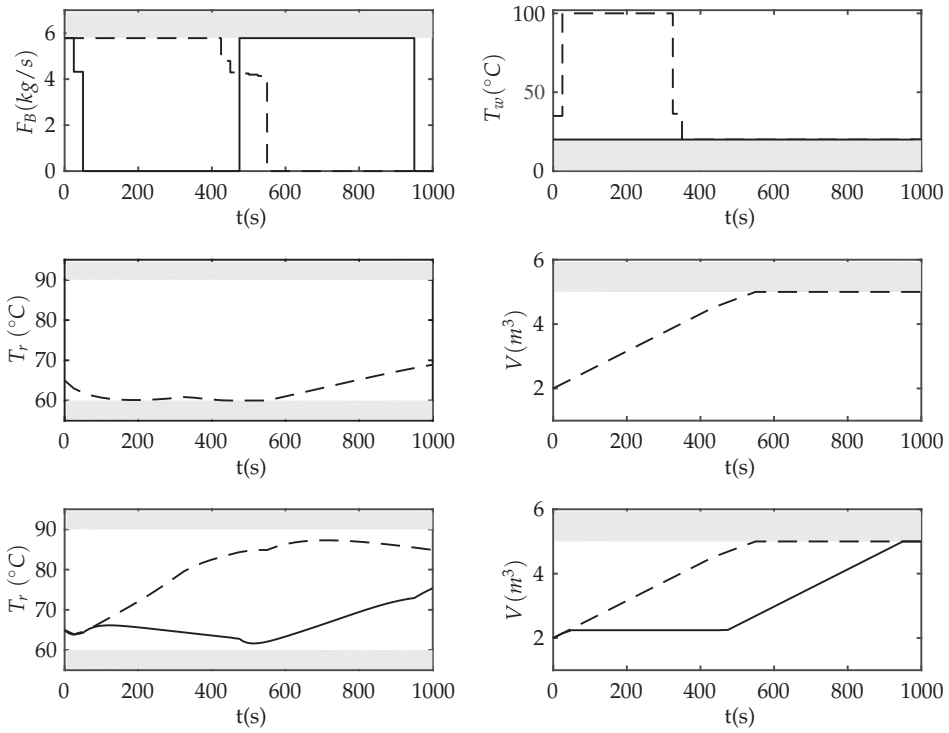


Figure 3.3 – Williams-Otto reactor. Shaded area: infeasible region. **Top plots:** *Input variables*: dashed lines: model optimal solution; solid lines: plant optimal solution. **Middle plots:** *Constrained output variables*: model optimal solution. **Bottom plots:** *Constrained output variables*: dashed lines: plant at the model optimal solution; solid lines: plant at the plant optimal solution.

values of the activation energies b_1 and b_2 and the pre-exponential factors a_1 and a_2. The parameter values and their uncertainty ranges are given in Table 3.2. The optimal input profiles obtained by solving Problem (3.35) for both the nominal model and the plant are shown in the top plots of Figure 3.3. As seen in the figure, the model

Table 3.2 – Williams-Otto reactor: Plant-model mismatch.

Parameter	Plant value	Nominal model value	Uncertainty range	Probability distribution
b_1 (K)	6000	6666.7	[5334, 8000]	uniform
b_2 (K)	8333.3	8750	[7500, 9166]	uniform
a_1 (s^{-1})	$1.6599 \cdot 10^6$	$1.8259 \cdot 10^6$	$[1.4109 \cdot 10^6, 1.9089 \cdot 10^6]$	uniform
a_2 (s^{-1})	$7.2117 \cdot 10^8$	$6.8511 \cdot 10^8$	$[6.1299 \cdot 10^8, 8.2935 \cdot 10^8]$	uniform

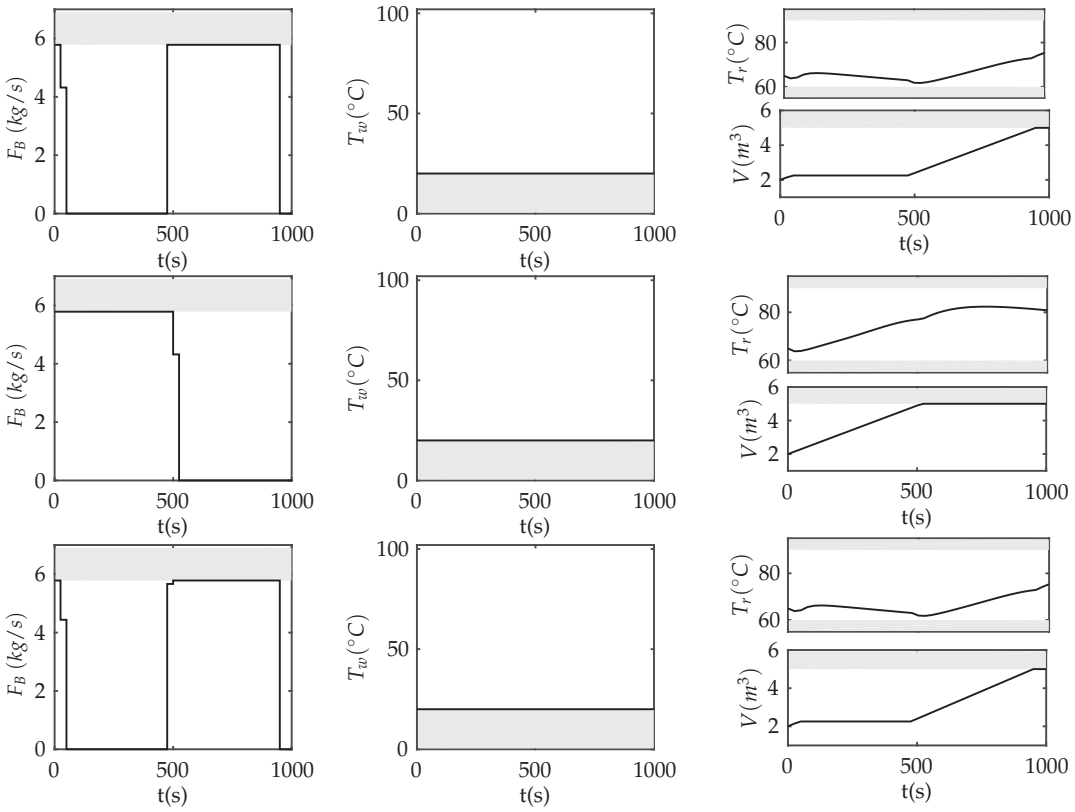


Figure 3.4 – Williams-Otto reactor. Plant input and output profiles upon convergence with different RTO methods. **Top plots:** MA. **Middle plots:** DMA. **Bottom plots:** ADMA.

and plant solutions are quite different. The optimal revenue for the plant is $3.14 \cdot 10^6$. However, upon applying the model solution to the plant, a sub-optimal revenue of $1.66 \cdot 10^6$ is obtained.

The optimal input profiles for the plant are assumed to be unknown. Hence, the goal of the different RTO methods is to improve upon the sub-optimal revenue resulting from applying the model optimal solution. The best RTO method is the one that requires the minimal experimental effort to reach plant optimality.

At first, MA with full gradient estimates is implemented. The top plots in Figure 3.4 show the input and output profiles obtained with MA upon convergence. Although MA starts from the input sequences given by the model solution, it is able to identify the correct set of constraints that are active at the plant optimum, thereby reaching the maximal possible revenue. However, MA requires full gradient estimation and, thus, incurs a large experimental cost at each RTO iteration, which makes the application of this type of MA prohibitive in practice.

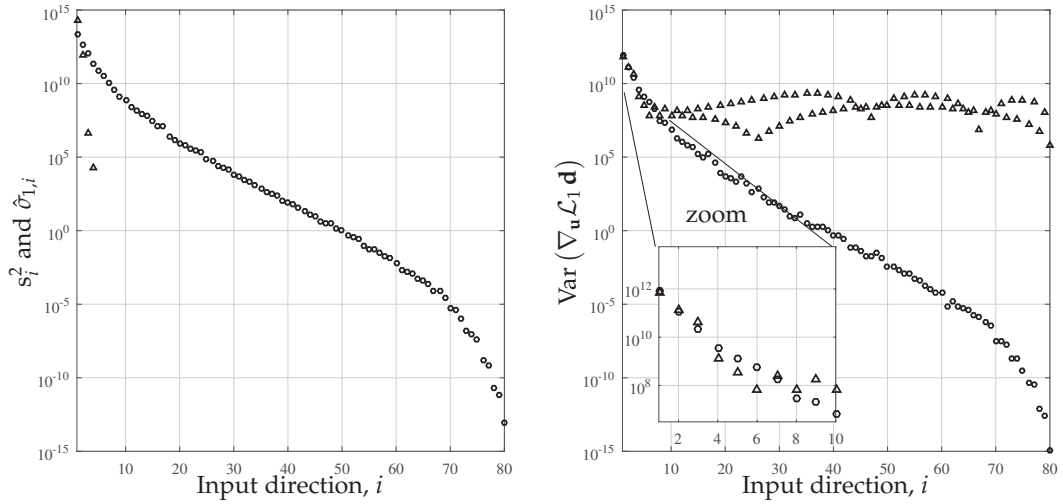


Figure 3.5 – Williams-Otto reactor: Comparison of the sensitivity matrices $\boldsymbol{A}^\star$ and $\hat{\boldsymbol{A}}_1$. (a) **Triangle**, $\triangle$ **:** Squared singular values of $\boldsymbol{A}^\star$. **Circle**, $\circ$ **:** Eigenvalues of $\hat{\boldsymbol{A}}_1$. (b) **Triangle**, $\triangle$ **:** $\mathbf{d} = \mathbf{w}_i$, the left singular vectors of $\boldsymbol{A}^\star$. **Circle**, $\circ$ **:** $\mathbf{d} = \hat{\mathbf{w}}_{1,i}$, the eigenvectors of $\hat{\boldsymbol{A}}_1$.

To implement the DMA and ADMA algorithms, the sensitivity matrices $\boldsymbol{A}^\star$ and $\hat{\boldsymbol{A}}_k$ are constructed. The number of Monte-Carlo samples for constructing $\hat{\boldsymbol{A}}_k$ is $N = 200$. The number of privileged directions is determined successively based on Criteria 1 and 2 in (3.32) and (3.33). To this end, the values of $n_{\max}$ and $v_{\min}$ are fixed at 4 and 1, respectively.

Criterion 1. The squared singular values of $\boldsymbol{A}^\star$ and the eigenvalues of $\hat{\boldsymbol{A}}_1$ at the model solution at $k = 1$ are plotted in Figure 3.5a. Notice that there is a large gap between the second and third singular values of $\boldsymbol{A}^\star$. Hence, based on Condition (3.6) or (3.32), the number of privileged directions for DMA can be fixed at $n_{\mathrm{r}} = 2$. However, for the same gap, the number of privileged directions with $\hat{\boldsymbol{A}}_1$ is more than $n_{\max} = 4$. Hence, for a fair comparison between DMA and ADMA, we fix the number of privileged directions for both algorithms at 4.

Criterion 2. The number of privileged directions can also be determined on the basis of variance values as described by (3.33). The variances associated with both the left singular vectors of $\boldsymbol{A}^\star$ and the eigenvalues of $\hat{\boldsymbol{A}}_1$ are plotted in Figure 3.5b. It is seen that the variances associated with the left singular vectors of $\boldsymbol{A}^\star$ do not exhibit a monotonic decrease except for the first few left singular vectors. In contrast, the variances associated with the eigenvectors of $\hat{\boldsymbol{A}}_1$ show a monotonic decrease. The median and the minimal value of the variance for $\hat{\boldsymbol{A}}_1$ are 0.51 and $1.1 \cdot 10^{-15}$, respectively. In comparison, the median and the minimal values of the variance for $\boldsymbol{A}^\star$ are $2.3 \cdot 10^8$ and $6.7 \cdot 10^5$, respectively. Hence, the eigenvectors of $\hat{\boldsymbol{A}}_1$ give a much better orthog-

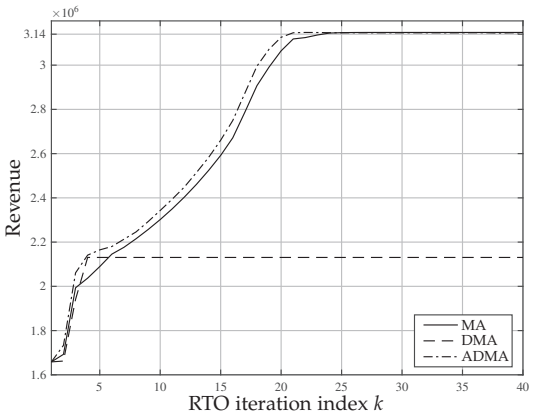


Figure 3.6 – Willliams-Otto reactor: Revenue evolution of the plant with different RTO methods.

onal decomposition of the input space, which is able to capture most of the global sensitivity in a relatively low number of directions. The threshold $v_{\min} = 1$ is not very useful here as it retains all 80 directions for DMA and 40 directions for ADMA. Hence, Criterion 2 also gives $n_r = 4$ for both DMA and ADMA. In the ADMA algorithm, we then keep the number of privileged directions fixed at 4 for every RTO iteration for the sake of comparison with DMA. The sets of 4 privileged directions computed by DMA and ADMA at the first RTO iteration are different as the model Lagrangian is a highly nonlinear function of the model parameters b_1 and b_2. Note that the 4 privileged directions in ADMA change from one iteration to the next due to the re-computation of $\hat{\boldsymbol{A}}_k$ at each iteration.

Upon application of DMA Algorithm 3.1, the plant input and output profiles reached upon convergence are shown in the middle plots of Figure 3.4. Although DMA successfully finds the optimal water temperature profile, it is unable to find the optimal profile for the feedrate of B. Obviously, adapting the gradients in the 4 privileged directions found by DMA is not sufficient to reach plant optimality as the gradient uncertainty along these directions is not sufficiently representative. The bottom plots of Figure 3.4 show the input and output profiles obtained with the ADMA Algorithm 3.3. As seen, ADMA successfully reaches the plant optimal profiles (see also Figure 3.6). This indicates that most of the gradient errors lie along the 4 privileged directions of ADMA.

The comparison of the different MA-based RTO methods is summarized in Table 3.3. As MA requires full gradient estimation, one must have 80 batches to evaluate the plant gradients at each RTO iteration. That amounts to 22.23 hours of waiting time per RTO iteration. This experimental time is reduced by applying DMA, which requires directional derivatives to be computed in only 4 directions, thus, needing only 4 batches, which reduces the experimental cost to 1.12 hours. However, the max-

Table 3.3 – Williams-Otto reactor: Comparison of different RTO methods.

RTO method	Per RTO iteration			Revenue ($\cdot 10^6$)
	No. of privileged directions, n_r	Avg. computational time of sensitivity matrix	Experimental cost	
MA	80	–	22.23 h	3.14
DMA	4	0 s ($A^\star$ computed only once offline)	1.12 h	2.13
ADMA	4	90.92 s ($\hat{A}_k$ computed via Algorithm 3.2)	1.12 h	3.14

imal revenue reached by DMA is only $2.13 \cdot 10^6$. ADMA, on the other hand, gives an optimal revenue of $3.14 \cdot 10^6$, while incurring the same experimental cost as DMA. This increase in revenue is made possible by the matrix $\hat{A}_k$ that requires on average a computational time of 90.92 s per RTO iteration when computed via Algorithm 3.2.[3]

3.5.2 Diketene-Pyrrole Reaction System

Next, the performances of different RTO methods are compared on the run-to-run optimization of a semi-batch reactor given in [109] or [22]. The reaction system is the acetoacetylation of pyrrole with diketene and consists of following reactions:

$$\begin{aligned} A + B &\xrightarrow{k_1} C, \\ 2B &\xrightarrow{k_2} D, \\ B &\xrightarrow{k_3} E, \\ C + B &\xrightarrow{k_4} F. \end{aligned}$$

The model equations, the initial conditions and the concentration of B in the feed used in this simulation study are given in A.2. The objective is to maximize the yield of product C, while penalizing large changes in the feedrate F_B of reactant B. The

[3]Simulations were conducted on a MacBook Pro with 2.5 GHz intel Core i7 processor. The software used is CasADi [4] version 3.2.3 in MATLAB version R2016a.

optimization problem can be written mathematically as

$$\max_{F_B(t)} \quad c_C(t_f)V(t_f) - \omega \int_0^{t_f} F_B^2(t)\,\mathrm{d}t \tag{3.36a}$$

$$\text{s.t.} \quad \text{model equations (A.2)} \tag{3.36b}$$

$$c_B(t_f) \leq c_B^{\max} \tag{3.36c}$$

$$c_D(t_f) \leq c_D^{\max} \tag{3.36d}$$

$$0 \leq F_B(t) \leq F_B^{\max}. \tag{3.36e}$$

The values of the final batch time t_f, the maximal inlet flowrate $F_B^{\max}$, the maximal concentrations of species B and D at final time and the value of the weight ω are given in Table A.2. The problem is formulated as an NLP by using a piecewise-constant discretization of the input $F_B(t)$ comprised of 50 control stages. Hence, the input dimension is $n_u = 50$.

Structural plant-model mismatch is introduced by ignoring the third and fourth reactions in the model, that is, by taking $k_3 = 0$ and $k_4 = 0$ for the model. Also, it is assumed that the model parameters k_1 and k_2 are uncertain and uniformly distributed within the ranges corresponding to $\pm 20\%$ of the nominal values. The mismatch considered and the uncertainty ranges are given in Table 3.4. The optimal solutions for the model and the plant are shown in the top plot of Figure 3.7. The two input profiles are quite different from each other. The evolution of the model $c_B(t)$ and $c_D(t)$ at the model optimal solution is shown in the middle plots of Figure 3.7. The bottom plots of Figure 3.7 show the evolution of the plant $c_B(t)$ and $c_D(t)$ obtained upon application of both the model and the plant optimal solutions. It is observed that the terminal constraint on the concentration of reactant B is not at its upper limit for the plant when the model optimal solution is applied. The model optimal solution applied to the plant result in a sub-optimal yield of 0.3865 moles, whereas the plant optimal yield is 0.5050 moles.

Table 3.4 – Kinetic parameters for Diketene-pyrrole reaction.

Parameter	Plant value	Nominal model value	Uncertainty range	Probability distribution
k_1 ($\mathrm{Lmol^{-1}min^{-1}}$)	0.053	0.053	[0.0424, 0.0636]	uniform
k_2 ($\mathrm{Lmol^{-1}min^{-1}}$)	0.128	0.128	[0.1024, 0.1536]	uniform
k_3 ($\mathrm{min^{-1}}$)	0.028	0	-	-
k_4 ($\mathrm{Lmol^{-1}min^{-1}}$)	0.001	0	-	-

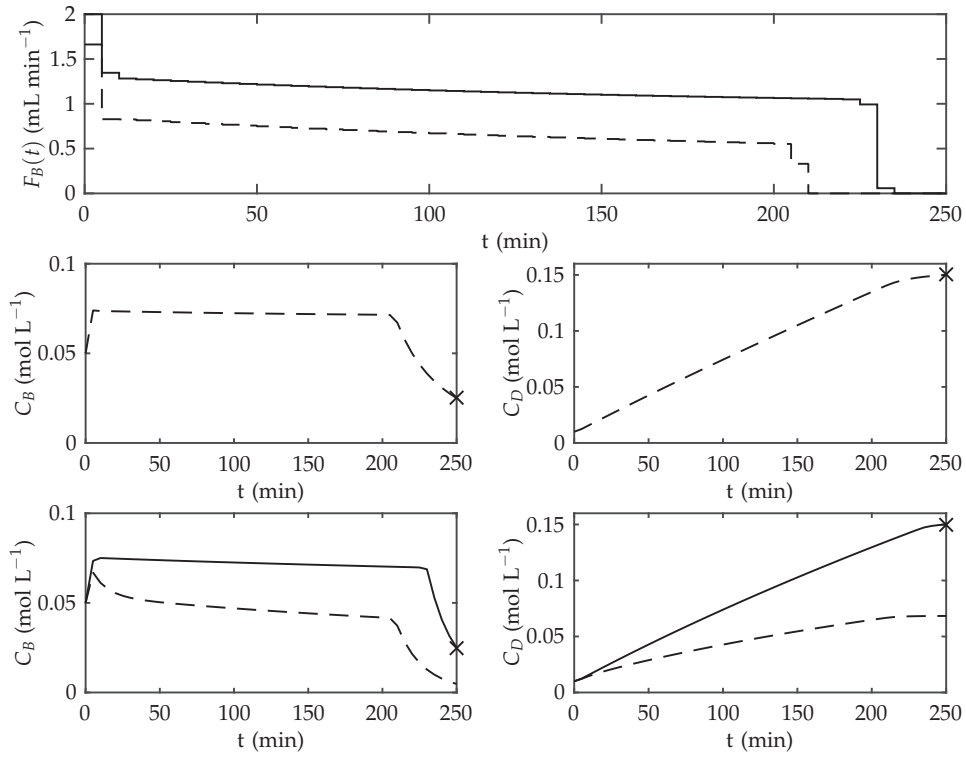


Figure 3.7 – Diketene-pyrrole reaction. **Top plots:** *Input profile*: dashed line: model optimal solution; solid line: plant optimal solution. **Middle plots:** *Output variables that are constrained at final time*: model optimal solution. **Bottom plots:** *Output variables that are constrained at final time*: dashed lines: plant response at the model optimal solution; solid lines: plant response at the plant solution. Cross, $\times$: terminal constraint threshold.

Since only two parameters k_1 and k_2 are uncertain, the local sensitivity matrix $\boldsymbol{A}^\star$ generates 2 privileged directions as per Condition (3.7). The variances along the left singular vectors of $\boldsymbol{A}^\star$ are plotted in Figure 3.8b. It is seen that the variances along remaining 48 directions do not decrease monotonically and, thus, more privileged directions cannot be selected. To construct $\hat{\boldsymbol{A}}_k$ via Algorithm 3.2, the number of Monte-Carlo samples is fixed at $N = 100$. Here, in contrast to the previous case study, we do not fix the number of privileged directions in ADMA; instead, we apply Criterion 1 in (3.32) at each RTO iteration by fixing $n_{\max} = 4$. The eigenvalues and variances computed at $k = 1$ are plotted in Figure 3.8. Criterion 1 gives 2 privileged directions at the first RTO iteration for ADMA. In this example, the two privileged directions found by DMA and ADMA are the same, which results from the fact that the model Lagrangian is only a weakly nonlinear function of the parameters k_1 and k_2. Note that these 2 privileged directions are *less privileged* at the next iterations since the privileged directions change with the input $F_B(t)$ from iteration to iteration. However,

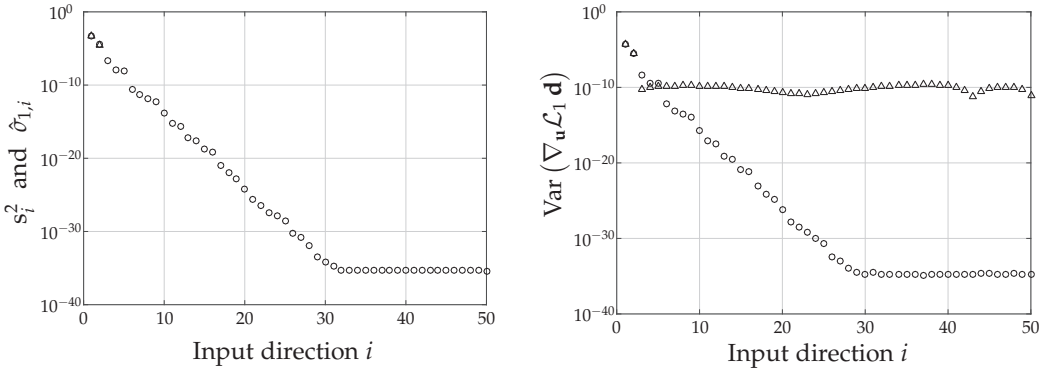


Figure 3.8 – Diketene-pyrrole reaction: Comparison of the sensitivity matrices $\boldsymbol{A}^\star$ and $\hat{\boldsymbol{A}}_1$. (a) **Triangle**, $\triangle$ **:** Squared singular values of $\boldsymbol{A}^\star$. **Circle**, $\circ$ **:** Eigenvalues of $\hat{\boldsymbol{A}}_1$. (b) **Triangle**, $\triangle$ **:** $\mathbf{d} = \mathbf{w}_i$, the left singular vectors of $\boldsymbol{A}^\star$. **Circle**, $\circ$ **:** $\mathbf{d} = \hat{\mathbf{w}}_{1,i}$, the eigenvectors of $\hat{\boldsymbol{A}}_1$.

since ADMA recomputes the privileged directions at each RTO iteration, it is able to always work with the most appropriate set of privileged directions. The number of privileged directions found at each RTO iteration using Criterion 1 is plotted in Figure 3.10b.

The input and output profiles reached upon convergence with MA, DMA and ADMA are shown in Figure 3.9. The evolution of the yield with the different RTO methods is shown in Figure 3.10a. Clearly, DMA exhibits a slight sub-optimality, whereas the MA and ADMA algorithms converge to plant optimality (at least as far as the yield value is concerned) despite the presence of significant plant-model mismatch.

The performance of the different RTO methods is compared in Table 3.5. MA with full gradient estimation reaches the optimal yield of 0.5050 moles at the large

Table 3.5 – Diketene-pyrrole reaction: Comparison of different RTO methods.

RTO method	Per RTO iteration			Yield (mol)
	No. of privileged directions, n_r	Avg. computational time of sensitivity matrix	Experimental cost	
MA	50	–	208.34 h	0.5050
DMA	2	0 s ($\boldsymbol{A}^\star$ computed only once offline)	8.34 h	0.5009
ADMA	2 to 3	5.43 s ($\hat{\boldsymbol{A}}_k$ computed via Algorithm 3.2)	8.34 to 12.5 h	0.5049

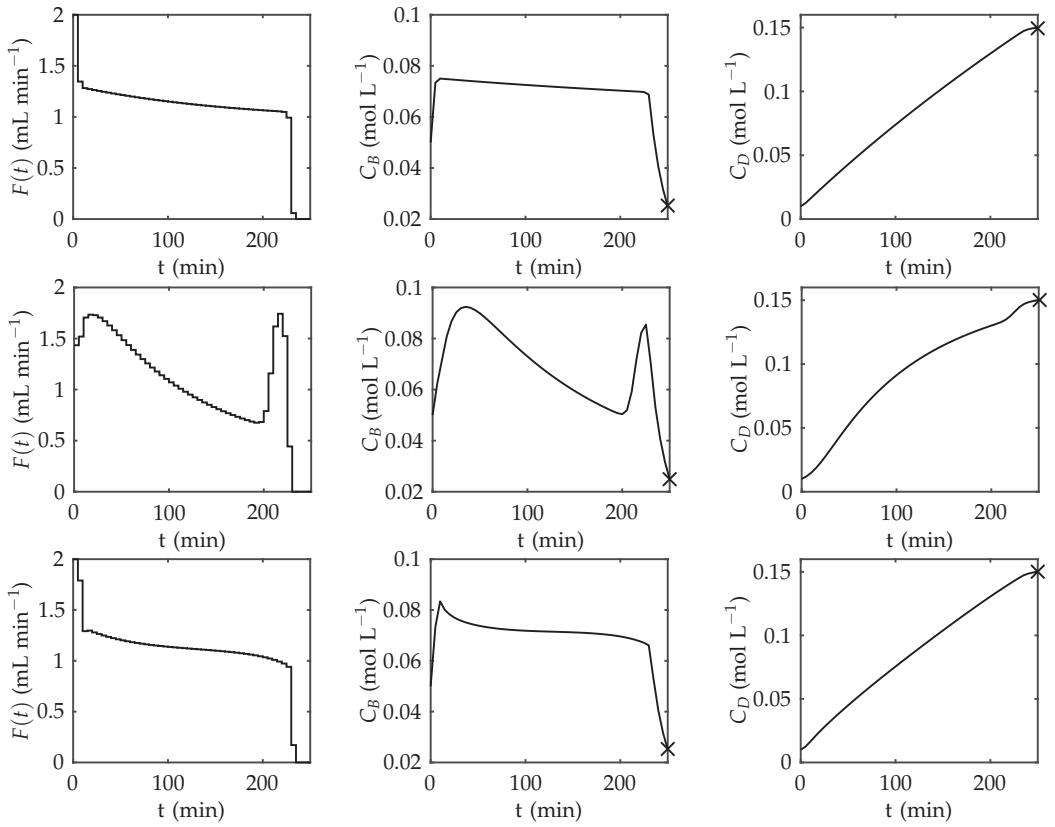


Figure 3.9 – Diketene-pyrrole reaction: Plant input and output profiles upon convergence with different RTO methods. **Top plots:** MA. **Middle plots:** DMA. **Bottom plots:** ADMA with $\hat{\boldsymbol{A}}_k$ computed via Algorithm 3.2.

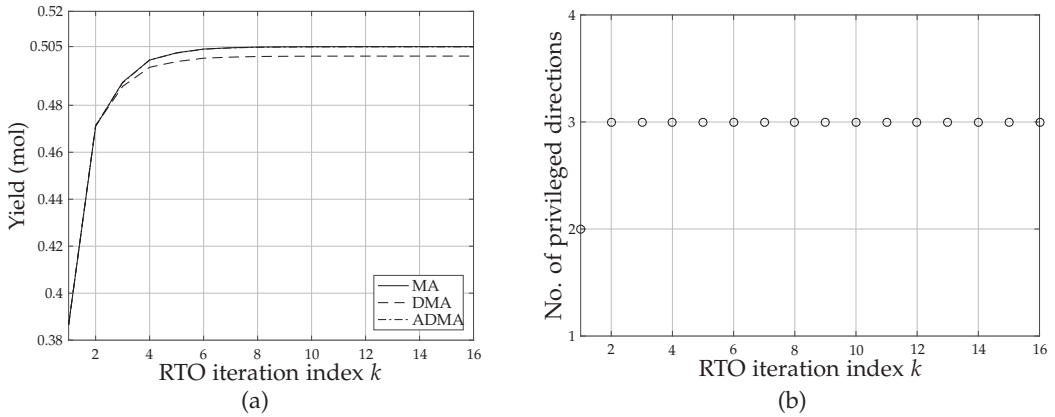


Figure 3.10 – Diketene-pyrrole reaction: Batch-to-batch evolution of the optimization strategy. (a) Yield evolution for different RTO methods. (b) Number of privileged directions at successive iterations.

experimental cost of 208.34 hours per RTO iteration. DMA significantly reduces the experimental cost to 8.34 hours per RTO iteration, but reaches a final yield of only 0.5009 moles. In comparison, ADMA nearly reaches optimal yield at an experimental cost of 8.34 to 12.5 hours. The average computational cost of $\hat{\boldsymbol{A}}_k$ is 5.43 seconds. ADMA gives the best performance as it comes very close to the plant optimum at a relatively low experimental cost. The computer and software used to perform the simulations are the same as in the previous case study.

3.6 Computational Aspects of Privileged Directions

Computation of the sensitivity matrix $\hat{\boldsymbol{A}}_k$ is an expensive task as it involves calculating the sensitivity of the model-Lagrangian, first with respect to inputs and then, with respect to the parameters, at each parameter sample. If the sensitivity computations are done via forward finite differences, then the computational cost of obtaining the privileged directions with $\hat{\boldsymbol{A}}_k$ is

$$\text{Number of model Lagrangian evaluations } = (n_{\mathrm{u}}+1)\times(n_{\theta}+1)\times N.$$

The above cost increases with increasing input and parameter dimensions. To reduce the computational costs, an alternative sensitivity matrix $\bar{\boldsymbol{A}}_k$ is proposed that requires only Lagrangian gradient evaluations at each parameter sample.

$$\bar{\boldsymbol{A}}_k = \int_{\Theta} \left(\nabla_{\mathbf{u}}\mathcal{L}_k - \bar{\mathbf{m}}\right)^{\mathsf{T}} \left(\nabla_{\mathbf{u}}\mathcal{L}_k - \bar{\mathbf{m}}\right) \rho \,\mathrm{d}\boldsymbol{\theta}, \tag{3.37a}$$

$$\text{with} \quad \nabla_{\mathbf{u}}\mathcal{L}_k := \nabla_{\mathbf{u}}\mathcal{L}(\mathbf{u}_k, \boldsymbol{\mu}_k, \boldsymbol{\theta}) \text{ and } \bar{\mathbf{m}} = \int_{\Theta} \left(\nabla_{\mathbf{u}}\mathcal{L}_k\right) \rho \,\mathrm{d}\boldsymbol{\theta} \tag{3.37b}$$

Under the assumption that the gradient vector $\nabla_{\mathbf{u}}\mathcal{L}(\mathbf{u},\boldsymbol{\theta}) := \frac{\partial \mathcal{L}(\mathbf{u},\boldsymbol{\theta})}{\partial \mathbf{u}} \in \mathbb{R}^{1\times n_{\mathrm{u}}}$ is bounded, that is,

$$||\nabla_{\mathbf{u}}\mathcal{L}(\mathbf{u},\boldsymbol{\theta})|| \leq L, \quad L>0 \qquad \forall \quad \boldsymbol{\mu} \in \mathbb{R}^{n_{\mathrm{g}}}_{\geq 0}, \quad \mathbf{u}\in\mathcal{U}, \quad \boldsymbol{\theta}\in\Theta,$$

where $||\cdot||$ is the Euclidean norm, then, the properties of $\bar{\boldsymbol{A}}_k$ are explored in the following lemma:

Lemma 3.4. *For all $\boldsymbol{\mu}_k \in \mathbb{R}^{n_{\mathrm{g}}}_{\geq 0}$ and for all $\mathbf{u}_k \in \mathcal{U}$, it holds that*

$$\int_{\Theta} \left(\left(\nabla_{\mathbf{u}}\mathcal{L}_k - \bar{\mathbf{m}}\right)\bar{\mathbf{w}}_{i,k} \right)^2 \rho \,\mathrm{d}\boldsymbol{\theta} = \bar{\sigma}_{i,k}, \quad i = 1,\ldots,n_{\mathrm{u}}, \tag{3.38}$$

where $\bar{\sigma}_{i,k}$ is the eigenvalue corresponding to the eigenvector $\bar{\mathbf{w}}_{i,k}$ of $\bar{\boldsymbol{A}}_k$.

Proof. The definition of $\bar{\sigma}_{i,k}$ implies that

$$\bar{\sigma}_{i,k} = \bar{\mathbf{w}}_{i,k}^{\mathsf{T}} \bar{A}_k \bar{\mathbf{w}}_{i,k}$$

which can be written as

$$\begin{aligned}
\bar{\sigma}_{i,k} &= \bar{\mathbf{w}}_{i,k}^{\mathsf{T}} \left(\int_{\Theta} \left(\nabla_{\mathbf{u}} \mathcal{L}_k - \bar{\mathbf{m}}\right)^{\mathsf{T}} \left(\nabla_{\mathbf{u}} \mathcal{L}_k - \bar{\mathbf{m}}\right) \rho \, \mathrm{d}\boldsymbol{\theta} \right) \bar{\mathbf{w}}_{i,k} \\
&= \int_{\Theta} \left(\left(\nabla_{\mathbf{u}} \mathcal{L}_k - \bar{\mathbf{m}}\right) \bar{\mathbf{w}}_{i,k} \right)^{\mathsf{T}} \left(\left(\nabla_{\mathbf{u}} \mathcal{L}_k - \bar{\mathbf{m}}\right) \bar{\mathbf{w}}_{i,k} \right) \rho \, \mathrm{d}\boldsymbol{\theta} \\
&= \int_{\Theta} \left(\left(\nabla_{\mathbf{u}} \mathcal{L}_k - \bar{\mathbf{m}}\right) \bar{\mathbf{w}}_{i,k} \right)^{2} \rho \, \mathrm{d}\boldsymbol{\theta}.
\end{aligned}$$

□

If an eigenvalue $\bar{\sigma}_{i,k} = 0$ then from the above lemma we have

$$\begin{aligned}
\left(\nabla_{\mathbf{u}} \mathcal{L}_k - \bar{\mathbf{m}}\right) \bar{\mathbf{w}}_{i,k} &= 0, \\
\left(\nabla_{\mathbf{u}} \mathcal{L}_k\right) \bar{\mathbf{w}}_{i,k} &= \bar{\mathbf{m}} \, \bar{\mathbf{w}}_{i,k} = \bar{\mathrm{c}} \in \mathbb{R}.
\end{aligned} \tag{3.39}$$

This implies that if an eigenvalue is zero then the parametric variations have no impact on the value of the model Lagrangian-gradient's projection along the corresponding eigenvector. Therefore, the model gradient is not need to be corrected along this eigenvector!

Lemma 3.4 has another interesting property that the eigenvalue $\bar{\sigma}_{i,k}$ is indeed the variance of the random quantity $\nabla_{\mathbf{u}} \mathcal{L}_k \, \bar{\mathbf{w}}_{i,k}$. Note that $\bar{\mathbf{m}}$ is the mean value of the Lagrangian gradient over the admissible parameter set Θ. Therefore, its projection on the eigenvector $\bar{\mathbf{w}}_{i,k}$ is the mean directional-derivative. Hence, the integral in (3.38) computes the variance of the directional derivative $\nabla_{\mathbf{u}} \mathcal{L}_k \, \bar{\mathbf{w}}_{i,k}$.

The matrix $\bar{\boldsymbol{W}}_k$ can be split into two submatrices, the matrix $\bar{\boldsymbol{W}}_{1,k} \in \mathbb{R}^{n_{\mathrm{u}} \times n_{\mathrm{r}}}$ and the matrix $\bar{\boldsymbol{W}}_{2,k} \in \mathbb{R}^{n_{\mathrm{u}} \times (n_{\mathrm{u}} - n_{\mathrm{r}})}$ as follows:

$$\bar{\boldsymbol{W}}_k = \left[\bar{\boldsymbol{W}}_{1,k} \; \bar{\boldsymbol{W}}_{2,k} \right], \tag{3.40a}$$

$$\bar{\boldsymbol{W}}_{1,k} = \left[\bar{\mathbf{w}}_{1,k} \cdots \bar{\mathbf{w}}_{n_{\mathrm{r}},k} \right] : \bar{\sigma}_{1,k} \geq \cdots \geq \bar{\sigma}_{n_{\mathrm{r}},k} > 0, \quad n_{\mathrm{r}} \leq n_{\mathrm{u}}, \tag{3.40b}$$

$$\bar{\boldsymbol{W}}_{2,k} = \left[\bar{\mathbf{w}}_{n_{\mathrm{r}}+1,k} \cdots \bar{\mathbf{w}}_{n_{\mathrm{u}},k} \right] : \bar{\sigma}_{n_{\mathrm{r}}+1,k} = \cdots = \bar{\sigma}_{n_{\mathrm{u}},k} = 0. \tag{3.40c}$$

3.6.1 Alternative Method for Computing Privileged Directions

With the help of Lemma 3.4, the privileged directions are found directly from the model Lagrangian gradient without explicitly computing its parametric sensitivity. Indeed, the eigenvalue decomposition of the matrix $\bar{\boldsymbol{A}}_k$ reveals the set of privileged directions as the eigenvectors of $\bar{\boldsymbol{A}}_k$ that constitutes $\bar{\boldsymbol{W}}_{1,k}$. As these eigenvectors have the

Algorithm 3.4 Computing approximation of $\bar{A}_k$ as $\hat{\bar{A}}_k$

Step 1: Draw N independent samples $\boldsymbol{\theta}_j$ from Θ using the probability density ρ.

Step 2: Compute the n_{u}-dimensional vector of the Lagrangian gradient at each sample:

$$\nabla_{\mathbf{u}}\mathcal{L}_k^{(j)} := \frac{\partial \mathcal{L}}{\partial \mathbf{u}}(\mathbf{u}_k, \boldsymbol{\mu}_k, \boldsymbol{\theta}_j), \quad j = 1, \ldots, N.$$

Step 3: Compute the mean of the sampled gradients

$$\hat{\bar{\mathbf{m}}} = \frac{1}{N}\sum_{j=1}^{N}\left(\nabla_{\mathbf{u}}\mathcal{L}_k^{(j)}\right).$$

Step 4: Compute $\hat{\bar{A}}_k$ as follows

$$\hat{\bar{A}}_k = \frac{1}{N}\sum_{j=1}^{N}\left(\nabla_{\mathbf{u}}\mathcal{L}_k^{(j)} - \hat{\bar{\mathbf{m}}}\right)^{\mathsf{T}}\left(\nabla_{\mathbf{u}}\mathcal{L}_k^{(j)} - \hat{\bar{\mathbf{m}}}\right). \tag{3.41}$$

Step 5: Compute the eigenvalue decomposition of matrix $\hat{\bar{A}}_k$ to obtain $\hat{\bar{W}}_k$

$$\hat{\bar{A}}_k = \hat{\bar{W}}_k \hat{\bar{\Sigma}}_k \hat{\bar{W}}_k^{\mathsf{T}}.$$

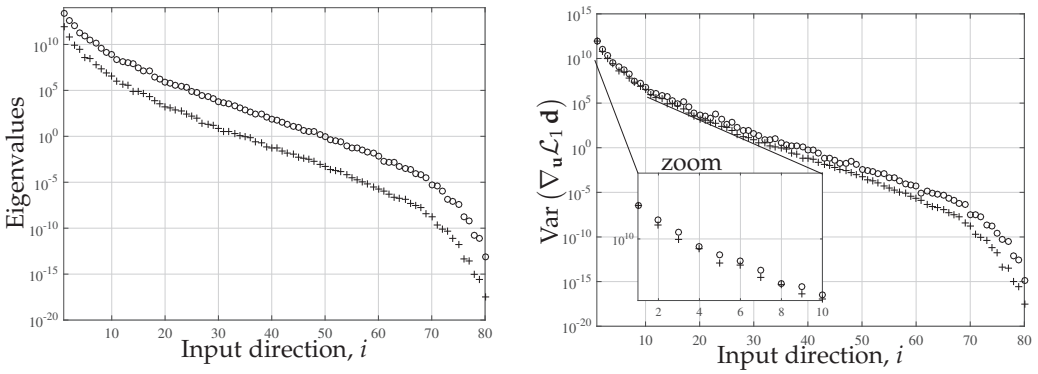
Step 6: Choose the privileged directions as the first n_{r} columns of the matrix $\hat{\bar{W}}_k$

$$\hat{\bar{W}}_{1,k} = [\hat{\bar{\mathbf{w}}}_{1,k} \cdots \hat{\bar{\mathbf{w}}}_{n_{\mathrm{r}},k}],$$
$$n_{\mathrm{r}} = \min\{i, n_{\max}\} : \hat{\bar{\sigma}}_{i+1,k} \ll \hat{\bar{\sigma}}_{i,k},$$

where $n_{\max}$ is the user defined upper bound on the number of privileged directions

largest eigenvalues (which is same as variance for $\bar{A}_k$), the gradient projections along them have the highest variability with respect to the parametric variations, whereas the gradient projections along the eigenvectors in $\bar{W}_{2,k}$ are insensitive to the parametric variations. The variance of the gradient projections along $\bar{W}_{2,k}$ is zero as eigenvalues are zero.

As the complexity of the process models may prohibit to analytically obtain the matrix $\bar{A}_k$, a procedure to approximate $\bar{A}_k$ is detailed in the Algorithm 3.4. Note that it is computationally much less expensive to approximate $\hat{\bar{A}}_k$ as only the derivative with respect to the input $\mathbf{u}$ is required to be computed at each sample.



(a) (b)

Figure 3.11 – Williams-Otto reactor: Comparison of the sensitivity matrices $\hat{\boldsymbol{A}}_1$ and $\hat{\hat{\boldsymbol{A}}}_1$. (a) **Circle**, $\circ$ **:** Eigenvalues of $\hat{\boldsymbol{A}}_1$. **Plus**, $+$ **:** Eigenvalues of $\hat{\hat{\boldsymbol{A}}}_1$. (b) **Circle**, $\circ$ **:** $\mathbf{d} = \hat{\mathbf{w}}_{1,i}$, the eigenvectors of $\hat{\boldsymbol{A}}_1$. **Plus**, $+$ **:** $\mathbf{d} = \hat{\hat{\mathbf{w}}}_{1,i}$, the eigenvectors of $\hat{\hat{\boldsymbol{A}}}_1$.

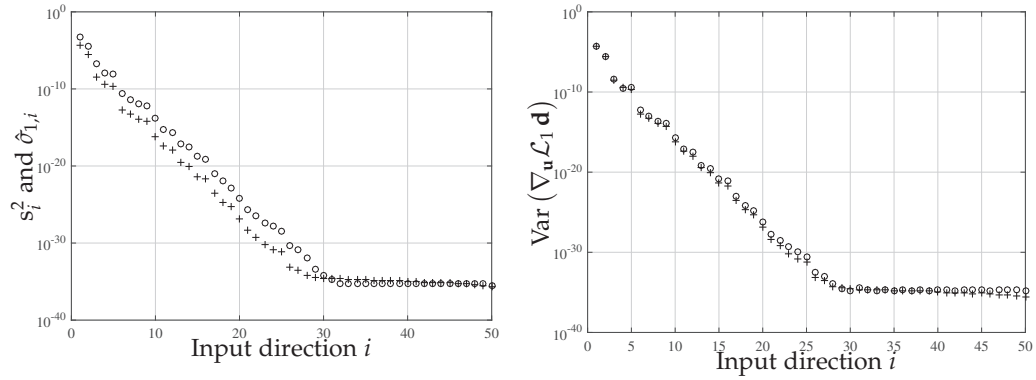
Case Studies Revisited

Performance of the two sensitivity matrices $\hat{\boldsymbol{A}}_k$ and $\hat{\hat{\boldsymbol{A}}}_k$ is compared in the simulation studies of the Williams-Otto reactor and Diketene-Pyrrole reaction system.

Williams-Otto semi-batch reactor: Proposed ADMA algorithm is applied to the semi-batch reactor with the privileged directions computed via $\hat{\hat{\boldsymbol{A}}}$. For comparison, the number of privileged directions is again fixed to 4. At 1^{st} RTO iteration, the eigenvalues and the variances computed for the two sensitivity matrices are plotted in the Figure 3.11. The variance plots show that the eigenvectors of the two sensitivity matrices have approximately the same variance distribution. The eigenvalue plots show that the eigenvectors for both the matrices are appropriately ranked. Note that the variance values in the right plot and the eigenvalues in the left plot are the same for $\hat{\hat{\boldsymbol{A}}}_k$. The application of ADMA algorithm with privileged directions obtained via $\hat{\hat{\boldsymbol{A}}}_k$ also lead to the plant-optimum. The ADMA performance using different sensitivity matrices is compared in the Table 3.6. Both sensitivity matrices lead to the similar performance

Table 3.6 – Williams-Otto reactor: Summary of different RTO methods.

RTO method	Per RTO iteration			Revenue ($\cdot 10^6$)
	No. of privileged directions, n_r	Avg. computation time of sensitivity matrix	Experimental cost	
ADMA with $\hat{\boldsymbol{A}}_k$ computed via Algo. 3.2	4	90.92 s	1.12 h	3.14
ADMA with $\hat{\hat{\boldsymbol{A}}}_k$ computed via Algo. 3.4	4	7.57 s	1.12 h	3.14



(a) (b)

Figure 3.12 – Diketene-pyrrole reaction: Comparison of the sensitivity matrices $\hat{\boldsymbol{A}}_1$ and $\hat{\tilde{\boldsymbol{A}}}_1$. (a) **Circle**, ∘ **:** Eigenvalues of $\hat{\boldsymbol{A}}_1$. **Plus**, $+$ **:** Eigenvalues of $\hat{\tilde{\boldsymbol{A}}}_1$. (b) **Circle**, ∘ **:** $\mathbf{d} = \hat{\mathbf{w}}_{1,i}$, the eigenvectors of $\hat{\boldsymbol{A}}_1$. **Plus**, $+$ **:** $\mathbf{d} = \hat{\tilde{\mathbf{w}}}_{1,i}$, the eigenvectors of $\hat{\tilde{\boldsymbol{A}}}_1$.

except for the computation time required to obtain the matrices. Computation of the sensitivity matrix $\hat{\tilde{\boldsymbol{A}}}_k$ takes only 7.57 seconds per RTO iteration as compared to 90.92 seconds consumed in the computation of $\hat{\boldsymbol{A}}_k$. The number of RTO iterations required to reach plant optimality with $\hat{\boldsymbol{A}}_k$ and with $\hat{\tilde{\boldsymbol{A}}}_k$ is 23 and 25, respectively.

Diketene-Pyrrole reaction system: At RTO iteration $k = 1$, the eigenvalues and variance of the gradient projection on the eigenvectors for the two sensitivity matrices are plotted in the Figure 3.12. The eigenvalue gap between the second and the third eigenvalues in the left plot of the figure reveals 2 privileged directions for both the matrices. Here too, the variance plot show that the variances are similarly distributed over the eigenvectors of $\hat{\boldsymbol{A}}_1$ and $\hat{\tilde{\boldsymbol{A}}}_1$. Results of applying ADMA in this study using different sensitivity matrices is compared in the Table 3.7. The ADMA algorithm performs similarly with either of the sensitivity matrices. The principle difference is in the computation time, $\hat{\tilde{\boldsymbol{A}}}_k$ consumes 0.55 seconds on average per RTO iteration whereas sensitivity matrix $\hat{\boldsymbol{A}}_k$ consumes 5.43 seconds. Both sensitivity matrices take 7 RTO iterations to reach plant optimality.

Table 3.7 – Diketene-Pyrrole reaction system: Summary of different RTO methods.

RTO method	Per RTO iteration			Yield (mol)
	No. of privileged directions, n_r	Avg. computation time of sensitivity matrix	Experimental cost	
ADMA with $\hat{\boldsymbol{A}}_k$ computed via Algo. 3.2	2 to 3	5.43 s	8.34 to 12.5 h	0.5049
ADMA with $\hat{\tilde{\boldsymbol{A}}}_k$ computed via Algo. 3.4	2 to 3	0.55 s	8.34 to 12.5 h	0.5049

3.6.2 A Second Alternative that Exploits Active Subspace

The computational expense of computing privileged directions is directly proportional to the dimension of the parameter space n_θ. The parameter space can be reduced by taking into account only those parameters that actually influence the model Lagrangian function (and thereby, its gradient in $\mathbf{u}$). The set of parameters that have no influence on the Lagrangian function can then be discarded for computing the privileged directions. If the number of influential parameters is considerably less than n_θ than the computational cost of obtaining privileged directions is reduced proportionally.

To divide the parameter space into influential and non-influential parameters, the concept of active subspaces is utilized. At a given $(\mathbf{u}_k, \boldsymbol{\mu}_k)$, the global sensitivity of the model Lagrangian can be approximated that results in a low-dimensional structure in the parameter space called active subspace. However, computing active subspace itself can be computationally intensive as it involves evaluating gradients of the Lagrangian function in the parameter space (i.e., $\nabla_\theta \mathcal{L}$) at each random sample of the parameter. Fortunately, [28] proposed a linearization technique that can reduce the computational cost of discovering active subspaces. To capture the gradient information, the primary interest is only in local behavior. Therefore, a local linear model can be fitted on a subset of random parameter samples. Instead of computing gradients directly, the fitted linear model provides the approximate gradient. Algorithm 3.5 computes the active subspaces by fitting local linear models. The choice of the sample sizes M and $\bar{N}$ in Algorithm 3.5 are detailed in [28]. The active subspace matrix $\hat{\boldsymbol{Q}}_{1,k}$ computed at $(\mathbf{u}_k, \boldsymbol{\mu}_k)$ leads to the influential parameter space $\Theta_r \subset \mathbb{R}^m$ by performing the following linear transformation

$$\boldsymbol{\theta}_r = (\hat{\boldsymbol{Q}}_{1,k})^\mathsf{T} \boldsymbol{\theta}, \quad \boldsymbol{\theta}_r \in \Theta_r, \quad \Theta_r = (\hat{\boldsymbol{Q}}_{1,k})^\mathsf{T} \Theta. \tag{3.42}$$

As $m \ll n_\theta$, computing privileged directions using Algorithm 3.2 with sampling only in the reduced space Θ_r. The sensitivity matrix in Step 2 of Algorithm 3.2 is calculated only with respect to $\boldsymbol{\theta}_r$. The overall Lagrangian evaluations required for obtaining the privileged directions using active subspaces would then be $\big((1+n_\mathrm{u}) \times (1+m) \times \bar{M}\big) + \bar{N}$, where $\bar{M} < N$. Note that as only m parameters are utilized for sensitivity computations, therefore, the total number of random samples required in Algorithm 3.2 reduces to $\bar{M}$.

Comparison of Different Computational Methods

There is a significant advantage of finding the privileged directions via the matrix $\hat{\hat{\boldsymbol{A}}}_k$ in Algorithm 3.4 as compared to finding the privileged directions via $\hat{\boldsymbol{A}}_k$ in Algorithm 3.2. It is computationally much less expensive to approximate $\hat{\hat{\boldsymbol{A}}}_k$ as only the derivative

Algorithm 3.5 Active subspace estimation with local linear models [28].

Step 1: Choose $M > \bar{N}$ and an integer n such that $\mathsf{n} < \bar{N}$.

Step 2: Draw $\bar{N}$ independent samples $\{\boldsymbol{\theta}_j\}$ from Θ according to the probability density ρ and compute $\mathcal{L}_k^{(j)} := \mathcal{L}(\mathbf{u}_k, \boldsymbol{\mu}_k, \boldsymbol{\theta}_j)$ for each sample.

Step 3: Draw M independent samples $\{\bar{\boldsymbol{\theta}}_l\}$ from Θ according to ρ.

Step 4: For each $\bar{\boldsymbol{\theta}}_l$, find the n points from the set $\{\boldsymbol{\theta}_j\}$ nearest to $\bar{\boldsymbol{\theta}}_l$; denote this set by $\bar{\Theta}_l$. Let $\mathfrak{L}_l$ be the subset of $\{\mathcal{L}_k^{(j)}\}$ that corresponds to the points in $\bar{\Theta}_l$.

Step 5: Use least squares to fit the coefficients a_l and $\boldsymbol{b}_l$ of a local linear regression model,

$$\mathcal{L}_k^{(j)} \approx \mathsf{a}_l + \boldsymbol{b}_l^{\mathsf{T}} \boldsymbol{\theta}_j, \quad \boldsymbol{\theta}_j \in \bar{\Theta}_l, \ \mathcal{L}_k^{(j)} \in \mathfrak{L}_l.$$

Step 6: Compute the corresponding sensitivity matrix $\hat{\boldsymbol{C}}_k$ and its eigenvalue decomposition

$$\hat{\boldsymbol{C}}_k = \frac{1}{M} \sum_{l=1}^{M} \boldsymbol{b}_l \, \boldsymbol{b}_l^{\mathsf{T}} = \hat{\boldsymbol{Q}}_k \, \hat{\boldsymbol{\Pi}}_k \, (\hat{\boldsymbol{Q}}_k)^{\mathsf{T}}, \ \hat{\boldsymbol{Q}}_k = [\hat{\mathbf{q}}_{1,k} \cdots \hat{\mathbf{q}}_{n_\theta,k}], \text{ and } \hat{\boldsymbol{\Pi}}_k = \operatorname{diag}(\hat{\pi}_{1,k}, \ldots, \hat{\pi}_{n_\theta,k}).$$

Step 7: Select the active subspace as the columns of the matrix $\hat{\boldsymbol{Q}}_{1,k}$ obtained by partitioning $\hat{\boldsymbol{Q}}_k$ as

$$\hat{\boldsymbol{Q}}_{1,k} = [\hat{\mathbf{q}}_{1,k} \cdots \hat{\mathbf{q}}_{m,k}], \quad \hat{\boldsymbol{Q}}_{2,k} = [\hat{\mathbf{q}}_{m+1,k} \cdots \hat{\mathbf{q}}_{n_\theta,k}], \quad m : \hat{\pi}_{m,k} \gg \hat{\pi}_{m+1,k}\,.$$

with respect to the input $\mathbf{u}$ is required to be computed at each sample.

Similarly, finding privileged directions by computing sensitivities only with respect to the active subspace also reduces the computational cost significantly as it reduces the number of parameters for sensitivity computation. The reduction of computational burden via active subspaces is demonstrated on the optimization of the fuel-cell system case study which is detailed in Chapter 5 (see Table 5.6).

If the sensitivity computations are done via forward finite differences, then the computational costs of obtaining the privileged directions via the three proposed methods are summarized in the Table 3.8.

Table 3.8 – Comparison of different computational methods if the sensitivities are computed via forward finite differences.

Method	Total evaluations of the model Lagrangian
Algorithm 3.2 with full parameter space	$(n_{\mathrm{u}}+1)\times(n_\theta+1)\times N$
Algorithm 3.4	$(n_{\mathrm{u}}+1)\times N$
Algorithm 3.2 with active subspaces via Algorithm 3.5	$\big((n_{\mathrm{u}}+1)\times(m+1)\times\bar{M}\big)+\bar{N},\ \bar{M}<N,\ m\ll n_\theta$

3.7 Summary

Plant gradient estimation is vital to the performance of explicit-RTO schemes such as MA. Plant gradient estimation with large input dimensions become challenging as excessive plant experiments are required when the plant is sequentially (and locally) perturbed in each input direction. This need for information-rich plant data at each RTO iteration renders the MA scheme prohibitive in practical applications.

To overcome this issue, it is proposed to rely more on the model predictions of the KKT conditions. To gain confidence in the model, parametric perturbations are made that single-out the most sensitive components in the model gradient. To this end, a local sensitivity analysis of the model-Lagrangian gradient can be performed that reveals a handful of input directions. These directions are called privileged directions. The model gradient along the privileged directions cannot be trusted and, thus, requires plant-based corrections. Consequently, the plant gradient is estimated only in the privileged-direction space, thereby reducing the experimental cost significantly.

However, the local sensitivity analysis gets erratic when the model gradient is a nonlinear function of the parameters. In this case, the local sensitivity analysis does not reveal the appropriate set of privileged directions. Moreover, the gradient is obviously a function of inputs and, thus, as the input value changes from one RTO iteration to the other, the gradient sensitivity also changes. Therefore, sensitivity analysis done at the initial input value does not hold at the successive iterations.

This chapter addresses the aforementioned issues. Here, it is proposed to find the set of privileged directions via a global sensitivity analysis that is inspired by the active subspace theory. This global sensitivity analysis is used to rank the input directions in terms of the sensitivity of the model gradient with respect to the large parametric variations, thus revealing the correct set of privileged directions. These privileged directions are such that the gradient projection on them has a relatively large variance with respect to the parametric perturbations. Moreover, the proposed

sensitivity analysis is performed at each RTO iteration, thereby improving the accuracy of the analysis.

The proposed improvements in discovering the privileged directions result in a novel RTO methodology called ADMA. The effectiveness of the ADMA scheme is demonstrated on the run-to-run optimization of two different semi-batch reactors. It is shown that ADMA offers a nice balance between experimental cost and achieved performance when compared to both, MA with full gradient estimation, and DMA based on local sensitivities.

In addition, this chapter address the important issue of computational complexity in obtaining the global sensitivity information. To reduce the computational burden of computing the global sensitivity, two alternatives to the initially introduced method in Algorithm 3.2 are proposed.

In the first alternative, the computational expense of finding privileged directions via global sensitivity analysis is reduced by computing only the model Lagrangian gradient $\nabla_{\mathrm{u}}\mathcal{L}$ at the random samples of model parameters. Unlike in Algorithm 3.2, the parametric sensitivity $\nabla_{\mathrm{u}\,\theta}\mathcal{L}$ of the sampled gradients is not computed. Instead, the parametric sensitivity $\nabla_{\mathrm{u}\,\theta}\mathcal{L}$ is approximated by subtracting the mean value from each gradient sample. The resulting privileged directions are such that they contribute the most to the variance of the model gradient. An interesting property of the method is that the variance of the directional derivative computed along the eigenvector is equal to its eigenvalue. The proposed alternative approach of computing global sensitivity has been tested on the simulation studies showing a considerable improvement in the computational time.

The second alternative considers the case when the parameter space is large. The computational effort of the sensitivity matrix $\nabla_{\mathrm{u}\,\theta}\mathcal{L}$ is directly proportional to the number of parameters. Therefore, reducing the parameter space directly impacts the computational time. To reduce the parameter space, the concept of active subspace is directly applied. Active subspace finds a low-dimensional structure in parameter space that causes the most variability in the output of interest. Therefore, the active subspace of the model Lagrangian is computed by fitting (computationally inexpensive) local linear models to the sampled data. The resulting algorithm drastically reduces computational effort required for discovering privileged directions.

4 Generalized Model Adaptation

In the previous chapter, the handles utilized to offset the identified model deficiencies are the modifiers. Modifiers are able to successfully match the model-plant KKT conditions since they are designed for the very same purpose. However, modifier corrections remain local in inputs and have no impact on the predictions of the original model. Moreover, measurement noise makes it difficult to obtain accurate plant gradient estimates [58, 80]. Since the first-order modifiers are linear in the plant gradients, the measurement noise impacts the numerical optimization via modifiers.

On the other hand, adapting the model parameters $\boldsymbol{\theta}$, instead of modifiers, has its own benefits. For instance, adjusting model parameters impact the model in the whole input space and not just locally. Moreover, parameters may favorably impact the curvature information, thereby potentially increasing the convergence rate to the optimum [2]. Also, through a simulation study, it is shown in [80] that the noise in process data can be better handled by adapting model parameters. In addition, adapting model parameters is strongly advocated when the model is expected to be structurally correct in the sense that there exists parameter values such that the model and the plant have matching outputs and gradients.

However, since process models are not tailored to predict the KKT conditions, there may not exist a one-to-one mapping between model parameters and constraint and gradient quantities. That is, one parameter may simultaneously influence multiple constraints and/or gradient components. Moreover, another parameter may have little to no impact on any of these quantities. Consequently, a model may be inadequate for independently meeting the KKT conditions. However, to the best of the author's knowledge, there has been only a little attempt [22] to establish whether a given process model provides such a one-to-one mapping; and if not, then what specific elements of the KKT conditions are the models able to affect by parametric adjustment.

The main contribution of this chapter is investigating and quantifying the interac-

tion between model parameters and KKT conditions, thereby revealing to which extent independent satisfaction of the KKT conditions is possible. To this end, an equivalence between the modifiers and the model parameters is established that results in the discovery of *mirror parameters*. Mirror parameters are the model parameters that *mimic* the role played by modifiers towards successful KKT matching between the model and the plant. Hence, the mirror parameters can be directly adapted instead of the modifiers. For the cases with too few mirror parameters for independent KKT matching, it is proposed to additionally rely on modifiers.

A further contribution of this chapter is in establishing synergies between privileged directions and model parameters. As privileged directions result from analyzing the impact of parametric perturbations, there may exist a subset of parameters that contribute most towards finding privileged directions. Consequently, the model gradients along privileged directions can be matched via the adjustment of dedicated parameters.

The chapter is organized as follows: Section 4.1 introduces the property of structural independence that enables one-to-one, KKT to modifier mapping in MA. This section then provides a procedure to analyze if the model parameters can provide the same flexibility. Finally, it proposes an RTO scheme for KKT matching. Section 4.2 proposes to compensate with modifiers when the models lack enough parameters for independent KKT matching. Section 4.3 proposes methods to match the KKT conditions only partially when the models lack sufficient parameters for complete KKT matching and the experimental cost of complete matching is considered too high. In such a case, it is proposed to match only the privileged KKT quantities via a combination of parameter and modifier adaptation. Section 4.4 illustrates the proposed concepts via a simulation steady of the Williams-Otto plant operated at steady state. Finally, Section 4.5 provides the summary.

4.1 Matching all KKTs via Parametric Adjustments

The Lagrangian gradient and the constraints of the model and the plant are required to be adapted to ensure reaching plant optimality. Therefore, the vector constituting these two quantities is defined as:

Definition 4.1 (KKT vector and KKT elements)**.** *The Lagrangian gradient and the constraint quantities that are required to be matched for the plant and the model are collected into a single vector called* KKT *vector. The* KKT *vector for the model reads:*

$$\boldsymbol{\mathcal{M}}_k(\boldsymbol{\theta}) = \left[\nabla_{\mathrm{u}}\mathcal{L}_k(\boldsymbol{\theta}),\, \mathbf{G}_k^{\mathsf{T}}(\boldsymbol{\theta})\right]^{\mathsf{T}} \in \mathbb{R}^{n_{\mathrm{u}}+n_{\mathrm{g}}}. \tag{4.1}$$

The elements $\mathcal{M}_{i,k}$, $i = 1, \ldots, n_{\mathrm{u}} + n_{\mathrm{g}}$, *of the* KKT *vector* $\boldsymbol{\mathcal{M}}_k$ *are called model* KKT *elements.*

Similarly, $\boldsymbol{\mathcal{M}}_{p,k}$ is defined as the plant KKT *vector and its elements $\mathcal{M}_{p,i,k}$ as the plant* KKT *elements.*

In the context of RTO, a highly desirable property of a model is its ability to satisfy the KKT conditions of the plant. That is, $\boldsymbol{\theta}$ can be adapted such that the plant and the model KKT conditions are the same. This can be achieved by enforcing the KKT matching via parameter adaptation as follows:

$$\boldsymbol{\theta}_{k+1} \in \underset{\boldsymbol{\theta}}{\operatorname{argmin}} \left|\left|\boldsymbol{\mathcal{M}}_k(\boldsymbol{\theta}) - \boldsymbol{\mathcal{M}}_{p,k}\right|\right|^2, \tag{4.2}$$

where $||\cdot||$ is the 2-norm. A model can successfully match all the elements in $\boldsymbol{\mathcal{M}}_{p,k}$ if it has the property of (global or local) structural correctness as defined below.

Definition 4.2 (Structurally correct model)**.** *In the context of* RTO*, a model is said to be* ***globally structurally correct*** *if there exists a $\boldsymbol{\theta}^\star \in \Theta$ such that the following conditions are satisfied*

$$\left(\frac{\partial \mathcal{L}}{\partial \mathbf{u}}\right)_{\boldsymbol{\theta}^\star} = \frac{\partial \mathcal{L}_p}{\partial \mathbf{u}} \quad \forall\, \mathbf{u} \in \mathcal{U} \subset \mathbb{R}^{n_\mathrm{u}}, \quad \boldsymbol{\mu} \in \mathbb{R}^{n_\mathrm{g}}. \tag{4.3a}$$

$$\mathbf{G}(\mathbf{u}, \boldsymbol{\theta}^\star) = \mathbf{G}_p(\mathbf{u}) \quad \forall\, \mathbf{u} \in \mathcal{U}. \tag{4.3b}$$

On the other hand, a model is said to be ***locally structurally correct*** *at $(\mathbf{u}_k, \boldsymbol{\mu}_k)$ if there exists $\boldsymbol{\theta}_{k+1} \in \mathbb{R}^{n_\theta}$ such that the following conditions are satisfied:*

$$\left(\nabla_\mathrm{u} \mathcal{L}_k\right)_{\boldsymbol{\theta}_{k+1}} = \nabla_\mathrm{u} \mathcal{L}_{p,k}, \tag{4.4a}$$

$$\left(\mathbf{G}_k\right)_{\boldsymbol{\theta}_{k+1}} = \mathbf{G}_{p,k}, \tag{4.4b}$$

$$\textit{or,} \quad \left(\boldsymbol{\mathcal{M}}_k\right)_{\boldsymbol{\theta}_{k+1}} = \boldsymbol{\mathcal{M}}_{p,k}. \tag{4.4c}$$

Global structural correctness guarantees that the model-based problem (2.2) and the plant problem (2.1) have essentially the same KKT point. However, the experimental cost of verifying global structural correctness is enormous, since this would require to evaluate the plant and its gradients in the entire input space.

On the other hand, local structural correctness is a highly desired property in iterative RTO schemes, where the parameters are adjusted so as to locally match the plant behavior. However, some parameters may be non-influential and, thus, unable to satisfy (4.4). Moreover, a parameter may be influential for more than one element of the KKT vector $\boldsymbol{\mathcal{M}}_k$ and, therefore, that KKT element can not be manipulated independently. Hence, one would like to examine whether adjusting model parameters can influence the $n_\mathrm{u} + n_\mathrm{g}$ KKT elements. To this end, the concept of structural independence of a model is introduced.

Remark 4.1 (Relation to model adequacy)**.** *In the context of* RTO*, model-adequacy require-*

ments have been described in [45] and discussed in detail in [82]. The so-called model-adequacy criterion is fulfilled by a process model if there exist values of the parameter $\boldsymbol{\theta}$ such that the model and the plant share the same local minimum $\mathbf{u}_p^\star$. This implies that, in addition to the KKT *conditions, the second-order necessary conditions of optimality must also be satisfied at $\mathbf{u}_p^\star$. However, a drawback of this criterion is that $\mathbf{u}_p^\star$ is unknown and, therefore, it cannot be verified a priori.*

In contrast, the property of global structural correctness guarantees that model parameter values exist such that the model-adequacy criterion is met up to the KKT *conditions, that is, excluding the second-order optimality conditions. However, model adequacy does not imply global structural correctness.*

When only KKT *conditions are considered, local structural correctness is the generalization of the model adequacy criterion since model adequacy is defined only for $\mathbf{u}_p^\star$. Also, model adequacy does not imply local structural correctness to be met for a given input except at $\mathbf{u}_p^\star$.*

4.1.1 Structural Independence and Mirror Parameters

To examine whether adjusting model parameters can *independently* influence the $n_\mathrm{u} + n_\mathrm{g}$ KKT elements, the concept of structural independence is introduced.

Definition 4.3 (Structurally independent model)**.** *In the context of* RTO *and assuming $n_\theta \geq n_\mathrm{u} + n_\mathrm{g}$, a model is said to be structurally independent at $(\mathbf{u}_k, \boldsymbol{\mu}_k)$ if there exists a bijective mapping $\boldsymbol{T}_k : \mathbb{R}^{n_\theta} \to \mathbb{R}^{n_\theta}$ such that the following conditions are satisfied:*

$$\left[\, \boldsymbol{\theta}^{\mathcal{L}},\, \boldsymbol{\theta}^{\mathrm{G}},\, \boldsymbol{\theta}^{c} \,\right] = \boldsymbol{T}_k\, \boldsymbol{\theta}, \quad \boldsymbol{\theta}^{\mathcal{L}} \in \mathbb{R}^{n_\mathrm{u}},\, \boldsymbol{\theta}^{\mathrm{G}} \in \mathbb{R}^{n_\mathrm{g}}. \tag{4.5a}$$

$$\left(\frac{\partial^2 \mathcal{L}}{\partial \boldsymbol{\theta}^{\mathcal{L}}\, \partial \mathbf{u}}\right)_{\mathbf{u}_k, \boldsymbol{\mu}_k} = \operatorname{diag}\left(f_k^{(1)}(\theta_1^{\mathcal{L}}),\, \ldots,\, f_k^{(n_\mathrm{u})}(\theta_{n_\mathrm{u}}^{\mathcal{L}})\right), \tag{4.5b}$$

$$\left(\frac{\partial^2 \mathcal{L}}{\partial \boldsymbol{\theta}^{\mathrm{G}}\, \partial \mathbf{u}}\right)_{\mathbf{u}_k, \boldsymbol{\mu}_k} = \mathbf{0}, \tag{4.5c}$$

$$\left(\frac{\partial \mathbf{G}}{\partial \boldsymbol{\theta}^{\mathrm{G}}}\right)_{\mathbf{u}_k} = \operatorname{diag}\left(\bar{f}_k^{(1)}(\theta_1^{\mathrm{G}}),\, \ldots,\, \bar{f}_k^{(n_\mathrm{g})}(\theta_{n_\mathrm{g}}^{\mathrm{G}})\right), \tag{4.5d}$$

$$\left(\frac{\partial \mathbf{G}}{\partial \boldsymbol{\theta}^{\mathcal{L}}}\right)_{\mathbf{u}_k} = \mathbf{0}, \tag{4.5e}$$

where $f_k^{(i)} : \mathbb{R} \to \mathbb{R}$, $\bar{f}_k^{(i)} : \mathbb{R} \to \mathbb{R}$ are scalar functions, θ_i^{G} and $\theta_i^{\mathcal{L}}$ are the i^{th} elements of the vectors $\boldsymbol{\theta}^{\mathrm{G}}$ and $\boldsymbol{\theta}^{\mathcal{L}}$, respectively.

In the above definition it is assumed that there exists values of $\boldsymbol{\theta}$ such that the diagonal matrices in (4.5b) and (4.5d) are full rank. The following theorem shows that structural independence is an inherent property of MA that enables it to locally match with the plant KKT elements.

Theorem 4.1 (In MA, structural independence ⇒ local structural correctness)**.** *The modified model* (2.16) *in* MA *is structurally independent at* $(\mathbf{u}_k, \boldsymbol{\mu}_k)$ *and, thus, locally structurally correct at* $(\mathbf{u}_k, \boldsymbol{\mu}_k)$*.*

Proof. Here the parameter $\boldsymbol{\theta}$ is replaced by the modifier vector $\left[\boldsymbol{\lambda}_k^{\mathcal{L}}, \boldsymbol{\varepsilon}_k^{\mathrm{G}}\right]$ and the bijective mapping $\boldsymbol{T}_k$ is taken as the identity matrix. Then, upon computing the sensitivity with respect to the modifiers of the modified model in (2.16), the following equations are obtained

$$\left(\frac{\partial^2 \mathcal{L}_{m,k}}{\partial \boldsymbol{\lambda}_k^{\mathcal{L}} \partial \mathbf{u}}\right)_{\mathbf{u}_k, \boldsymbol{\mu}_k} = \boldsymbol{I}_{n_{\mathrm{u}}}, \tag{4.6a}$$

$$\left(\frac{\partial^2 \mathcal{L}_{m,k}}{\partial \boldsymbol{\varepsilon}_k^{\mathrm{G}} \partial \mathbf{u}}\right)_{\mathbf{u}_k, \boldsymbol{\mu}_k} = \mathbf{0}, \tag{4.6b}$$

$$\frac{\partial \mathbf{G}_{m,k}}{\partial \boldsymbol{\lambda}_k^{\mathcal{L}}} = \mathbf{0}. \tag{4.6c}$$

$$\frac{\partial \mathbf{G}_{m,k}}{\partial \boldsymbol{\varepsilon}_k^{\mathrm{G}}} = \boldsymbol{I}_{n_{\mathrm{g}}}, \tag{4.6d}$$

where $\boldsymbol{I}_{n_{\mathrm{u}}}$ is the n_{u}-dimensional identity matrix and similarly, $\boldsymbol{I}_{n_{\mathrm{g}}}$ is the n_{g}-dimensional identity matrix. This implies that the modifiers in MA render the modified model structurally independent.

Integrating the sensitivity equations in (4.6) gives

$$\nabla_{\mathrm{u}} \mathcal{L}_{m,i,k} = \lambda_{i,k}^{\mathcal{L}} + \mathrm{c}_{i,k},\ \mathrm{c}_{i,k} \in \mathbb{R}, \quad i = 1, \ldots, n_{\mathrm{u}}, \tag{4.7a}$$

$$\mathrm{G}_{m,i,k} = \varepsilon_{i,k}^{\mathrm{G}} + \bar{\mathrm{c}}_{i,k},\ \bar{\mathrm{c}}_{i,k} \in \mathbb{R}, \quad i = 1, \ldots, n_{\mathrm{g}}, \tag{4.7b}$$

where $\nabla_{\mathrm{u}} \mathcal{L}_{m,i,k}$ and $\mathrm{G}_{m,i,k}$ are the i^{th} components of the vectors $\nabla_{\mathrm{u}} \mathcal{L}_{m,k}$ and $\mathbf{G}_{m,k}$, $\lambda_{i,k}^{\mathcal{L}}$ and $\varepsilon_{i,k}^{\mathrm{G}}$ are the i^{th} components of the vectors $\boldsymbol{\lambda}_k^{\mathcal{L}}$ and $\boldsymbol{\varepsilon}_k^{\mathrm{G}}$, and $\mathrm{c}_{i,k}$ and $\bar{\mathrm{c}}_{i,k}$ are constants. Obviously, there exist modifier values such that the plant and the modified model have locally matching Lagrangian gradient and constraints, that is,

$$\nabla_{\mathrm{u}} \mathcal{L}_{m,k} = \nabla_{\mathrm{u}} \mathcal{L}_{p,k} \text{ and } \mathbf{G}_{m,k} = \mathbf{G}_p(\mathbf{u}_k).$$

□

KKT matching can be achieved if the model offers sufficient degrees of freedom for adaptation. In MA, the added modifier terms provide the required flexibility. The $n_{\mathrm{u}} + n_{\mathrm{g}}$ modifiers in MA allows to independently manipulate the $n_{\mathrm{u}} + n_{\mathrm{g}}$ KKT elements. Modifiers are in fact tailored to enforce the KKT matching, since they make the modified model in (2.16) locally structurally correct. Moreover, in MA, the local-structural correctness is verified a priori, that is, modifiers ensure that the Lagrangian

gradient and the constraints of the model and the plant can be matched locally, no matter how different the plant and the model mappings are.

Since, the modified model in (2.16) is affine in modifiers, local sensitivities of the KKT conditions also represent the global sensitivities. The sensitivity equations (4.6) imply that each of the $n_u + n_g$ modifier elements affects only the corresponding KKT element and has no impact on the rest of the $n_u + n_g - 1$ KKT elements. This *decoupling* (facilitated by the modifiers) enables independent KKT matching. Hence, the modified model in (2.16) is an ideal candidate for RTO.

However, if such an independent structure is already present in a given model, then the modifiers are not needed. That is, one can attempt to adjust the parameter vectors $\boldsymbol{\theta}^{\mathcal{L}}$ and $\boldsymbol{\theta}^{\mathrm{G}}$ that offer the same independent KKT matching ability as modifiers. In this sense, the parameter vectors $\boldsymbol{\theta}^{\mathcal{L}}$ and $\boldsymbol{\theta}^{\mathrm{G}}$ resemble the modifier vectors $\boldsymbol{\lambda}_k^{\mathcal{L}}$ and $\boldsymbol{\varepsilon}_k^{\mathrm{G}}$.

Definition 4.4 (Mirror parameters)**.** *The parameter vectors $\boldsymbol{\theta}^{\mathcal{L}}$ and $\boldsymbol{\theta}^{\mathrm{G}}$ (provided they exist) are referred to as mirror parameters as they mirror the role of the modifiers $\boldsymbol{\lambda}_k^{\mathcal{L}}$ and $\boldsymbol{\varepsilon}_k^{\mathrm{G}}$ in enabling structural independence, thereby facilitating independent* KKT *matching.*

The main challenge lies in finding the mirror parameters. In the next subsection, an attempt is made to discover the parameter vectors $\boldsymbol{\theta}^{\mathcal{L}}$ and $\boldsymbol{\theta}^{\mathrm{G}}$ in a given model.

4.1.2 Discovering Mirror Parameters via Active Subspaces

Analyzing the structural independence of a given model requires computation of the mirror parameters. To this end, each KKT element $\mathcal{M}_{i,k}$ of the model is analyzed and its influential and non-influential parameter spaces $\boldsymbol{\mathcal{I}}_k^{(i)}$ and $\boldsymbol{\mathcal{NI}}_k^{(i)}$ are found via active subspaces as detailed in the Section 2.6.1.

Subsequently, any direction $\mathbf{d}_k^{(i)} \in \mathbb{R}^{n_\theta}$ in parameter space is able to independently match the corresponding KKT element $\mathcal{M}_{i,k}$, provided it belongs to the influential space of that KKT element and to the non-influential space of the rest of the KKT elements. Put differently,

$$\mathbf{d}_k^{(i)} \in \boldsymbol{\mathcal{D}}_k^{(i)} \subseteq \mathbb{R}^{n_\theta}, \quad i = 1, \ldots, (n_u + n_g), \tag{4.8a}$$

$$\text{with} \quad \boldsymbol{\mathcal{D}}_k^{(i)} := \boldsymbol{\mathcal{I}}_k^{(i)} \bigcap_{j=1}^{j=(n_u+n_g)} \boldsymbol{\mathcal{NI}}_k^{(j)}, \quad j \neq i. \tag{4.8b}$$

The direction $\mathbf{d}_k^{(i)} \in \boldsymbol{\mathcal{D}}_k^{(i)}$ provides a linear combination of the parameters $\boldsymbol{\theta}$ that can enforce independent KKT matching. Ideally, each KKT element should have such a dedicated parameter direction.

Theorem 4.2. *For a given model, if $\mathcal{D}_k^{(i)}$ is a non-empty set for $i = 1, \ldots, (n_u + n_g)$, then the model is structurally independent.*

Proof. Consider (2.18) and substitute $f = \mathcal{M}_{i,k}$. The resulting matrix is denoted as $C_k^{(i)}$ (the superscript i implies that the matrix corresponds to the i^{th} KKT element), which diagonalizes as:

$$C_k^{(i)} = Q_k^{(i)} \, \Pi_k^{(i)} \left(Q_k^{(i)}\right)^{\mathsf{T}}, \quad \Pi_k^{(i)} = \operatorname{diag}\left(\pi_{1,k}^{(i)}, \ldots, \pi_{n_\theta,k}^{(i)}\right),$$

with $\pi_{1,k}^{(i)} \geq \cdots \geq \pi_{n_\theta,k}^{(i)} \geq 0$; $Q_k^{(i)} \in \mathbb{R}^{n_\theta \times n_\theta}$ is an orthonormal matrix, whose columns $\mathbf{q}_{1,k}^{(i)}, \ldots, \mathbf{q}_{n_\theta,k}^{(i)}$ are the normalized eigenvectors of $C_k^{(i)}$. [1] Now $Q_k^{(i)}$ can be split as:

$$Q_k^{(i)} = \begin{bmatrix} Q_{1,k}^{(i)} & Q_{2,k}^{(i)} \end{bmatrix}, \quad Q_{1,k}^{(i)} \in \mathbb{R}^{n_\theta \times m}.$$

Define $\vartheta_{1,k}^{(i)} \in \mathbb{R}^m$ and $\vartheta_{2,k}^{(i)} \in \mathbb{R}^{n_\theta - m}$ as:

$$\vartheta_{1,k}^{(i)} := \left(Q_{1,k}^{(i)}\right)^{\mathsf{T}} \theta, \text{ and } \vartheta_{2,k}^{(i)} := \left(Q_{2,k}^{(i)}\right)^{\mathsf{T}} \theta.$$

Any $\theta \in \mathbb{R}^{n_\theta}$ can be expressed in terms of $\vartheta_{1,k}^{(i)}$ and $\vartheta_{2,k}^{(i)}$ as:

$$\theta = Q_k^{(i)} \left(Q_k^{(i)}\right)^{\mathsf{T}} \theta = Q_{1,k}^{(i)} \left(Q_{1,k}^{(i)}\right)^{\mathsf{T}} \theta + Q_{2,k}^{(i)} \left(Q_{2,k}^{(i)}\right)^{\mathsf{T}} \theta = Q_{1,k}^{(i)} \vartheta_{1,k}^{(i)} + Q_{2,k}^{(i)} \vartheta_{2,k}^{(i)}.$$

Hence, $\mathcal{M}_{i,k}(\theta)$ can be written as:

$$\mathcal{M}_{i,k}(\theta) = \mathcal{M}_{i,k}\left(Q_{1,k}^{(i)} \vartheta_{1,k}^{(i)} + Q_{2,k}^{(i)} \vartheta_{2,k}^{(i)}\right).$$

Assuming $\pi_{m+1,k}^{(i)} = \cdots = \pi_{n_\theta,k}^{(i)} = 0$, then, from Lemma 2.4 and Condition (2.24),

$$\nabla_{\vartheta_{2,k}^{(i)}} \mathcal{M}_{i,k}(\theta) = 0 \quad \forall\, \theta \in \Theta. \tag{4.9}$$

Hence,

$$\mathcal{M}_{i,k}(\theta) \approx \mathcal{M}_{i,k}\left(Q_{1,k}^{(i)} \vartheta_{1,k}^{(i)}\right) = \mathsf{f}_k^{(i)}\left(\vartheta_{1,k}^{(i)}\right),$$

where $\mathsf{f}_k^{(i)}$ is a scalar function.

Recall that $\mathbf{d}_k^{(i)} \in \mathcal{I}_k^{(i)} := \operatorname{col}\left(Q_{1,k}^{(i)}\right)$. Hence, fixing the rotated parameters resulting

[1] Note that the subscript k indicates that the matrix $C_k^{(i)}$, and thus also the quantities derived from it, are defined for the specific values of $(\mathbf{u}_k, \mu_k)$ realized at the k^{th} RTO iteration. Hence, as with k the values of $\mathbf{u}_k$ and μ_k change, the matrix $C_k^{(i)}$ and the quantities derived from it also change.

from directions other than $\mathbf{d}_k^{(i)}$ in $Q_{1,k}^{(i)}$ to their nominal value gives:

$$\mathcal{M}_{i,k}(\boldsymbol{\theta}) \approx \bar{\mathsf{f}}_k^{(i)}\left(\theta_k^{(i)}\right), \quad \theta_k^{(i)} := \left(\mathbf{d}_k^{(i)}\right)^{\mathsf{T}}\boldsymbol{\theta}. \tag{4.10}$$

Moreover, since $\mathbf{d}_k^{(i)} \in \boldsymbol{\mathcal{NI}}_k^{(l)} := \mathrm{col}\left(Q_{2,k}^{(l)}\right)\ \forall\, l \in \{1,\ldots,n_{\mathrm{u}}+n_{\mathrm{g}}\},\ l \neq i$. Therefore,

$$\nabla_{\theta_k^{(i)}}\mathcal{M}_{l,k}(\boldsymbol{\theta}) = 0 \quad \forall\, \boldsymbol{\theta} \in \Theta,\ l \in \{1,\ldots,n_{\mathrm{u}}+n_{\mathrm{g}}\},\ l \neq i. \tag{4.11}$$

Note that (4.9), (4.10) and (4.11) hold for all $i \in \{1,\ldots,n_{\mathrm{u}}+n_{\mathrm{g}}\}$. Hence, the model is structurally independent. □

The scalar component $\theta_k^{(\mathrm{i})}$ is such that it affects only the corresponding i$^{\mathrm{th}}$ KKT element and not the remaining KKT elements. These components, for $i = 1,\ldots,(n_{\mathrm{u}}+n_{\mathrm{g}})$, can be used to construct the parameter vectors $\boldsymbol{\theta}^{\mathrm{G}}$ and $\boldsymbol{\theta}^{\mathcal{L}}$ as follows:

$$\boldsymbol{\theta}^{\mathcal{L}} = \left[\theta_k^{(1)}\cdots\theta_k^{(n_{\mathrm{u}})}\right] = \left[\mathbf{d}_k^{(1)}\ \cdots\ \mathbf{d}_k^{(n_{\mathrm{u}})}\right]^{\mathsf{T}}\boldsymbol{\theta}, \quad \mathbf{d}_k^{(i)} \in \boldsymbol{\mathcal{D}}_k^{(i)}, \quad i = 1,\ldots,n_{\mathrm{u}}. \tag{4.12a}$$

$$\begin{aligned}\boldsymbol{\theta}^{\mathrm{G}} &= \left[\theta_k^{(n_{\mathrm{u}}+1)}\cdots\theta_k^{(n_{\mathrm{u}}+n_{\mathrm{g}})}\right] \\ &= \left[\mathbf{d}_k^{(n_{\mathrm{u}}+1)}\ \cdots\ \mathbf{d}_k^{(n_{\mathrm{u}}+n_{\mathrm{g}})}\right]^{\mathsf{T}}\boldsymbol{\theta}, \quad \mathbf{d}_k^{(i)} \in \boldsymbol{\mathcal{D}}_k^{(i)}, \quad i = n_{\mathrm{u}}+1,\ldots,(n_{\mathrm{u}}+n_{\mathrm{g}}).\end{aligned} \tag{4.12b}$$

Remark 4.2. *The non-influential parameter space is not necessarily formed by the eigenvectors with zero eigenvalues. Hence, parametric perturbations along the direction* $\mathbf{d}_k^{(i)} \in \boldsymbol{\mathcal{D}}_k^{(i)}$ *may still affect the* KKT *elements other than* $\mathcal{M}_{i,k}$*. However, since the eigenvalues corresponding to the non-influential parameter spaces are required to be relatively small, this influence is negligible. Moreover, some directions may not strictly satisfy Condition* (4.8)*. These directions may still be used to form mirror parameters if the Euclidean norm of their projections on each of the column spaces in* (4.8b) *is relatively larger than their projection on the respective complementary column spaces.*

4.1.3 Proposed RTO Scheme for Complete KKT Matching

The parameters $\boldsymbol{\theta}^{\mathrm{G}}$ and $\boldsymbol{\theta}^{\mathcal{L}}$ can be adjusted iteratively to correct the model in order to reach the plant KKT conditions. The resulting RTO scheme consists of three steps: first, the model-based optimization problem is solved; second, the input vector is filtered; and third, at the obtained solution, the parameters $\boldsymbol{\theta}^{\mathrm{G}}$ and $\boldsymbol{\theta}^{\mathcal{L}}$ are adjusted so as to enforce KKT matching. At the k^{th} RTO iteration, the optimization step takes the following form:

$$\textit{Optimization step}: \quad \left(\mathbf{u}_{k+1}^{\star}, \boldsymbol{\mu}_{k+1}^{\star}\right) = \underset{\mathbf{u}}{\mathrm{argmin}} \quad \Phi(\mathbf{u}, \boldsymbol{\theta}_k) \tag{4.13a}$$

$$\text{s.t.} \quad \mathrm{G}_i(\mathbf{u}, \boldsymbol{\theta}_k) \leq 0, \quad i = 1,\ldots,n_{\mathrm{g}}. \tag{4.13b}$$

The resulting RTO input is filtered before being applied to the plant:

$$\textit{Input update :} \quad \mathbf{u}_{k+1} := (\boldsymbol{I} - \boldsymbol{K})\,\mathbf{u}_k + \boldsymbol{K}\mathbf{u}^{\star}_{k+1}, \tag{4.14a}$$

$$\boldsymbol{\mu}_{k+1} := \boldsymbol{\mu}^{\star}_{k+1}, \tag{4.14b}$$

where $\boldsymbol{K} \in \mathbb{R}^{n_\mathrm{u} \times n_\mathrm{u}}$ is a diagonal gain matrix with diagonal elements in the interval $(0, 1]$.

To adjust the parameters, first the influential and non-influential parameter spaces are found for each KKT element using Algorithm 2.1. Then, the directions $\mathbf{d}_k^{(i)}$, $i = 1, \ldots, n_\mathrm{u} + n_\mathrm{g}$ are chosen such that they satisfy (4.8). The unused $(n_\theta - (n_\mathrm{u} + n_\mathrm{g}))$ rotated parameters that constitute the parameter vector $\boldsymbol{\theta}^c_k$, result from the parameter subspace that is the orthogonal complement to the subspace of directions $\mathbf{d}_k^{(i)}$. These parameters are fixed at $\boldsymbol{\theta}_k$. The following transformation describes the aforementioned step:

$$\left[\, \bar{\boldsymbol{\theta}}^{\mathcal{L}}_k,\, \bar{\boldsymbol{\theta}}^{\mathrm{G}}_k,\, \boldsymbol{\theta}^c_k \,\right] = \boldsymbol{T}_k\, \boldsymbol{\theta}_k, \tag{4.15a}$$

$$\boldsymbol{T}_k = \left[\, \mathbf{d}_k^{(1)},\, \mathbf{d}_k^{(2)} \,\cdots\, \mathbf{d}_k^{(n_\mathrm{u}+n_\mathrm{g})},\, \boldsymbol{D}_k \,\right]^{\mathsf{T}}, \tag{4.15b}$$

where $\boldsymbol{D}_k \in \mathbb{R}^{n_\theta \times (n_\theta - n_\mathrm{u} - n_\mathrm{g})}$ is the orthogonal complement to the column space of the directions $\mathbf{d}_k^{(i)}$, $i = 1, \ldots, n_\mathrm{u} + n_\mathrm{g}$, and $\boldsymbol{\theta}^c_k$ contains the unused parameters. At the k^{th} RTO iteration, the parameters $\boldsymbol{\theta}^{\mathcal{L}}$ and $\boldsymbol{\theta}^{\mathrm{G}}$ satisfying (4.12) act as the decision variables of the following parameter estimation step:

$$\textit{Parameter estimation step :} \quad \left(\boldsymbol{\theta}^{\mathcal{L}}_{k+1},\, \boldsymbol{\theta}^{\mathrm{G}}_{k+1}\right) = \underset{\boldsymbol{\theta}^{\mathcal{L}},\, \boldsymbol{\theta}^{\mathrm{G}}}{\operatorname{argmin}} \quad \left|\left|\boldsymbol{\mathcal{M}}_{k+1} - \boldsymbol{\mathcal{M}}_{p,k+1}\right|\right|^2, \tag{4.16a}$$

$$\left|\boldsymbol{\theta}^{\mathcal{L}} - \bar{\boldsymbol{\theta}}^{\mathcal{L}}_k\right| \leq \Delta^{\mathcal{L}}, \quad \left|\boldsymbol{\theta}^{\mathrm{G}} - \bar{\boldsymbol{\theta}}^{\mathrm{G}}_k\right| \leq \Delta^{\mathrm{G}}, \tag{4.16b}$$

where $\Delta^{\mathcal{L}} \subset \mathbb{R}^{n_\mathrm{u}}$ and $\Delta^{\mathrm{G}} \subset \mathbb{R}^{n_\mathrm{g}}$ restrict the parameter values around the current values and avoid oscillations from one RTO iteration to the next. The updated parameters read:

$$\boldsymbol{\theta}_{k+1} = (\boldsymbol{T}_k)^{-1} \left[\, \boldsymbol{\theta}^{\mathcal{L}}_{k+1},\, \boldsymbol{\theta}^{\mathrm{G}}_{k+1}, \boldsymbol{\theta}^c_k \,\right]. \tag{4.17}$$

4.2 Synergies between Parameters and Modifiers

Although structural independence is highly desirable, many models are not structurally independent as illustrated by the following examples.

Example 4.1. *Consider the unconstrained model-based optimization problem*

$$\mathcal{L}(\mathbf{u}, \boldsymbol{\theta}) = \theta_1 u_1 u_2 + \theta_2 \quad \textit{with } \mathbf{u} = [u_1,\, u_2]^\mathsf{T} \textit{ and } \boldsymbol{\theta} = [\theta_1,\, \theta_2]^\mathsf{T}. \tag{4.18}$$

The sensitivity equation for this model is

$$\frac{\partial^2 \mathcal{L}}{\partial \boldsymbol{\theta}\, \partial \mathbf{u}} = \begin{bmatrix} u_2 & 0 \\ u_1 & 0 \end{bmatrix} \tag{4.19}$$

The first column corresponding to θ_1 contains more than one non-zero element. Consequently, fixing the value of θ_1 fixes both components of the Lagrangian gradient. Hence, the two gradient components cannot be manipulated independently. Moreover, θ_2 has no influence on any of the two gradient components.

Example 4.2. *Now consider the following unconstrained model-based optimization problem*

$$\mathcal{L}(\mathbf{u}, \boldsymbol{\theta}) = \left(\frac{\theta_1}{\theta_2}\right) u_1 + u_2 \;\; \textit{with } \mathbf{u} = [u_1,\, u_2]^\mathsf{T} \textit{ and } \boldsymbol{\theta} = [\theta_1,\, \theta_2]^\mathsf{T},\; 2 \leq \theta_i \leq 3,\; i = 1, 2. \tag{4.20}$$

The corresponding sensitivity equation reads

$$\frac{\partial^2 \mathcal{L}}{\partial \boldsymbol{\theta}\, \partial \mathbf{u}} = \begin{bmatrix} \frac{1}{\theta_2} & \frac{-\theta_1}{\theta_2^2} \\ 0 & 0 \end{bmatrix}. \tag{4.21}$$

*The first row of the matrix has more than one non-zero element. Hence, there is more than one parameter that influences the first gradient component of the Lagrangian. In such a case, one of the (*less influential*) parameters can be fixed and the other is adjusted to match the corresponding* KKT *element. However, the second gradient component is not influenced by any of the parameters. Hence, the corresponding* KKT *element cannot be matched by adjusting the existing model parameters.*[2]

As demonstrated above, models may lack structural independence, thus making a direction set $\boldsymbol{\mathcal{D}}_k^{(i)}$ empty. If there are one or more such KKT elements that cannot be independently matched by adjusting the parameters, then a modifier component should be considered as a means of compensation. This way, the resulting partially modified model turns out to be structurally independent. This phenomenon is illustrated by the following example.

Example 4.3 (Examples 4.1 and 4.2 revisited)**.** *In Example 4.1, the two* KKT *elements have the same influential parameter space $[1, 0]^\mathsf{T}$. Hence, the direction sets $\boldsymbol{\mathcal{D}}_k^{(1)}$ and $\boldsymbol{\mathcal{D}}_k^{(2)}$ corresponding to the two* KKT *elements are empty. In such a case, we can still pair anyone of*

[2]The models used in the two examples are not structurally independent. However, depending on the corresponding plant behavior, they may still be globally or locally structurally correct.

the two KKT *elements with the influential parameter space and the remaining* KKT *element can then be paired with a modifier, thus enabling independent matching. The modified model would then take the following form:*

$$\mathcal{L}(\mathbf{u}, \boldsymbol{\theta}) = \theta_1 u_1 u_2 + \theta_2 + \lambda^{\mathcal{L}}_{1,k}\, u_1. \tag{4.22}$$

Although still structurally dependent, the above modified model is able to match the two KKT *elements independently. Here, the gradient component corresponding to u_2 can be matched by adjusting $[1, 0]^\mathsf{T}\boldsymbol{\theta}$, that is, θ_1. Although, the gradient component corresponding to u_1 is also influenced by θ_1, it can still be independently matched by adjusting the modifier $\lambda^{\mathcal{L}}_{1,k}$.*

In Example 4.2, the active subspaces can be found by assuming that both components of $\boldsymbol{\theta}$ are uniformly distributed between 2 and 3. The influential parameter space corresponding to the gradient component along u_1 is $[-0.69,\, 0.72]^\mathsf{T}$, whereas the gradient component along u_2 has no influential parameter space. Clearly, the gradient component along u_1 can be matched by adjusting its corresponding influential parameter space that satisfies (4.8)*. For the other gradient component, a modifier term can be added for independent matching, thereby making the modified model structurally independent:*

$$\mathcal{L}(\mathbf{u}, \boldsymbol{\theta}) = \left(\frac{\theta_1}{\theta_2}\right) u_1 + u_2 + \lambda^{\mathcal{L}}_{2,k}\, u_2 \tag{4.23}$$

Remark 4.3 (Is a gap in the spectrum of $\boldsymbol{C}_k^{(i)}$ necessary?)**.** *The eigenvalue gap for discovering the active subspace of the* KKT *elements are important if one is interested in finding a low-dimensional structure in parameter space that is most influential for the corresponding* KKT *element. However, in the context of matching all the* KKT *elements, presence of a low-dimensional structure is not necessarily of interest. The lack of such a gap in eigenvalues implies that either all parameters are influential or non-influential. If all the parameters are influential for a given* KKT *element then that element has a lot of potential directions available for its matching and if all the parameters are non-influential then that* KKT *element can be matched by the corresponding modifier. In both cases, the full matching of the* KKT *vector is minimally impacted by the presence or absence of the eigenvalue gap.*

4.3 Matching only Privileged KKT Elements

For safety and quality issues, satisfying process constraints is key to the performance of any RTO scheme. Therefore, it becomes essential to match at least the constraints. This requires measuring all the KKT elements of the vector $\boldsymbol{\mathcal{M}}_{p,k}$ that correspond to the plant constraints $\mathbf{G}_p$. Hence, the constraints remain as the privileged KKT elements. Constraints are fairly inexpensive to obtain as the direct output measurements of the plant running at $\mathbf{u}_k$ are sufficient to estimate the their values.

However, estimating the Lagrangian gradient of the plant is an expensive task. For

instance, the use of finite differences requires that the plant be locally perturbed at $\mathbf{u}_k$ in all input directions, each time waiting for the plant to settle at steady state. Hence, the experimental cost of obtaining the plant gradient per RTO iteration is ($n_\mathrm{u} \times$ process settling time).

One possible way of reducing the experimental cost is by relying to a greater extent on a process model. Consequently, in this section, the assumption that all KKT elements must be matched in order to reach optimality is challenged, and the possibility of a model being locally structurally correct without being fully structurally independent is explored. In addition to constraint matching, the goal is to find the most sensitive KKT elements corresponding to the Lagrangian gradient and then matching only them instead of the full Lagrangian gradient.

4.3.1 Constraint Matching

Certain process systems are such that plant optimality is driven by the active constraints [20, 22, 117] and matching only the constraint values is sufficient to reach near-optimal performance. In such a case, matching the KKT elements corresponding to the Lagrangian gradient are not considered and, consequently, the structural independence reduces to the following conditions:

$$\left(\frac{\partial \mathbf{G}}{\partial \boldsymbol{\theta}^\mathrm{G}}\right)_{\mathbf{u}_k} = \mathrm{diag}\left(\bar{f}_k^{(1)}(\theta_1^\mathrm{G}), \ldots, \bar{f}_k^{(n_\mathrm{g})}(\theta_{n_\mathrm{g}}^\mathrm{G})\right), \tag{4.24a}$$

$$\left(\frac{\partial \Phi}{\partial \boldsymbol{\theta}^\mathrm{G}}\right)_{\mathbf{u}_k} = \mathbf{0}. \tag{4.24b}$$

For a given model, parameters fulfilling the above conditions exist if the following set is non-empty:

$$\boldsymbol{\mathcal{D}}_k^{(\mathrm{G}_i)} := \boldsymbol{\mathcal{I}}_k^{(\mathrm{G}_i)} \bigcap \boldsymbol{\mathcal{NI}}_k^{(\Phi)} \bigcap_{j=1}^{j=n_\mathrm{g}} \boldsymbol{\mathcal{NI}}_k^{(\mathrm{G}_j)}, \quad j \neq i, \tag{4.25}$$

where $\boldsymbol{\mathcal{NI}}_k^{(\Phi)}$ is the non-influential parameter space of the cost function Φ; $\boldsymbol{\mathcal{I}}_k^{(\mathrm{G}_i)}$ and $\boldsymbol{\mathcal{NI}}_k^{(\mathrm{G}_i)}$ are the influential and non-influential parameter spaces of the constraint function G_i, respectively.

A direction $\mathbf{d}_k^{(\mathrm{G}_i)} \in \boldsymbol{\mathcal{D}}_k^{(\mathrm{G}_i)}$ can be used to compute the parameter $\theta_{i,k}^\mathrm{G}$ that matches the constraint G_i. If there are constraints that have no influential parameters, then such constraints can be matched by adapting zeroth-order modifiers. Such an RTO scheme has been tested on a batch-to-batch optimization problem in [117].

4.3.2 Pairing Privileged Directions with Influential Parameters

Since the Lagrangian gradient is considered along n_r privileged directions, only n_r parameters need to be available for adjustment. The most influential parameters are computed with the active subspaces. The pairing can be understood by revisiting Example 3.1. The directional derivative along the privileged direction $[1,\,1]^\mathsf{T}$ can be matched by adjusting the parameters along the influential parameter direction $[2,\,1,\,0]^\mathsf{T}$. As gradient matching along $[-1,\,1]^\mathsf{T}$ is not considered, a single finite-difference experiment is required at $\mathbf{u}_1$.

It might happen that (i) certain process constraints do not have an influential parameter space, or (ii) a significant overlap exists between the influential parameter spaces of the constraints and directional derivatives. Since the plant directional derivatives are typically expensive to obtain, it makes sense to want to meet the directional derivatives with their influential parameters and the constraints with zeroth-order modifiers.

Example 4.4 (Example 3.1 revisited)**.** *A constrained version of* (3.3) *is considered next:*

$$\mathcal{L}(\mathbf{u},\boldsymbol{\mu},\boldsymbol{\theta}) = (2\,\theta_1+\theta_2)\,u_1u_2+\theta_3+\mu_1\,(\theta_1+3\,\theta_2)\,u_1, \tag{4.26a}$$

$$\mathbf{G}(\mathbf{u},\boldsymbol{\theta}) = (\theta_1+3\,\theta_2)\,u_1 \leq 0. \tag{4.26b}$$

$$1 \leq u_i \leq 2, \quad i=1,2. \tag{4.26c}$$

The privileged direction $[1,\,1]^\mathsf{T}$ *has the influential parameter direction* $[2,\,1,\,0]^\mathsf{T}$*, while the constraint has the influential parameter direction* $[1,\,3,\,0]^\mathsf{T}$*. The two parameter directions are not orthogonal and, in fact, exhibit a significant overlap. Hence, the directional derivative along the privileged direction is matched via parameter adaptation along the corresponding influential parameter direction, while the constraint is matched by adapting the corresponding zeroth-order modifier* $\varepsilon^\mathsf{G}_{i,k}$*.*

The RTO scheme proposed in Section 4.1.3 is hence altered for the previous example. In other words, the optimization step of the proposed RTO scheme additionally has the modifier term $\boldsymbol{\varepsilon}^\mathsf{G}_k$ for the constraint. The overall scheme takes the following form:

$$\begin{aligned}\textit{Optimization step}: \quad \left(\mathbf{u}^\star_{k+1},\boldsymbol{\mu}^\star_{k+1}\right) = \underset{\mathbf{u}}{\operatorname{argmin}} \quad & (2\,\theta_{1,k}+\theta_{2,k})\,u_1u_2+\theta_{3,k},\\ & \boldsymbol{\theta}_k = [\theta_{1,k}\,\theta_{2,k}\,\theta_{3,k}]^\mathsf{T},\\ \text{s.t.} \quad & (\theta_{1,k}+3\,\theta_{2,k})\,u_1+\varepsilon^\mathsf{G}_{1,k} \leq 0.\end{aligned}$$

$$\begin{aligned}\textit{Input update}: \quad \mathbf{u}_{k+1} &:= (\boldsymbol{I}-\boldsymbol{K})\,\mathbf{u}_k+\boldsymbol{K}\mathbf{u}^\star_{k+1},\\ \boldsymbol{\mu}_{k+1} &:= \boldsymbol{\mu}^\star_{k+1},\end{aligned}$$

followed by a parameter estimation step,

$$\textit{Parameter estimation step}: \quad (\boldsymbol{\theta}_{k+1}^{\text{priv}}) = \underset{\boldsymbol{\theta}^{\text{priv}}}{\operatorname{argmin}} \quad ||\nabla_{\hat{W}_{1,k+1}} \mathcal{L}_{k+1} - \nabla_{\hat{W}_{1,k+1}} \mathcal{L}_{p,k+1}||^2,$$
$$\text{with} \quad \nabla_{\hat{W}_{1,k+1}} \mathcal{L}_{k+1} := \left. \frac{\partial \mathcal{L}\big((\mathbf{u}_{k+1} + \hat{\boldsymbol{W}}_{1,k+1}\,\mathbf{r}),\, \boldsymbol{\mu}_{k+1},\, \boldsymbol{\theta}_k\big)}{\partial \mathbf{r}} \right|_{\mathbf{r}=\mathbf{0}},$$

where $\nabla_{\hat{W}_{1,k+1}} \mathcal{L}_{k+1} \in \mathbb{R}^{1 \times n_{\mathrm{r}}}$ is the directional directive along the directions given by the columns of the matrix $\hat{\boldsymbol{W}}_{1,k+1}$, and $\mathbf{r} \in \mathbb{R}^{n_{\mathrm{r}}}$. The plant directional derivative $\nabla_{\hat{W}_{1,k+1}} \mathcal{L}_{p,k+1}$ is defined in a similar fashion. The matrix $\hat{\boldsymbol{W}}_{1,k+1}$ is computed via Algorithm 3.2. Similar to the parameters $\boldsymbol{\theta}^{\mathcal{L}}$ and $\boldsymbol{\theta}^{\mathsf{G}}$, the parameter $\boldsymbol{\theta}^{\text{priv}} \in \mathbb{R}^{n_{\mathrm{r}}}$ is obtained by computing the influential and non-influential spaces for each of the component of the directional-directive $\nabla_{\hat{W}_{1,k+1}} \mathcal{L}_{k+1}$.

For $k = 0$, assume that $\mathbf{u}_1 = [1, 1]^{\mathsf{T}}$ and $\mu_1 = 0$ are the values obtained after filtering, then the privileged direction matrix is $\hat{\boldsymbol{W}}_{1,1} = [1,\, 1]^{\mathsf{T}}$ and the corresponding parameter $\boldsymbol{\theta}^{\text{priv}}$ is $\boldsymbol{\theta}^{\text{priv}} = [2,\, 1,\, 0]^{\mathsf{T}}\boldsymbol{\theta}$. The two orthogonal parameter directions are fixed at $\boldsymbol{\theta}_0$.

4.4 Simulation Study

4.4.1 Williams-Otto Process

The proposed approach is applied to the process shown in Figure 4.1, which is a modification of the Williams-Otto plant proposed by [130]. The Williams-Otto process serves as a benchmark simulation to test RTO methods [57, 86, 94].

In the process shown in Figure 4.1, F_A and F_B are the fresh feeds of Species A and B, while F_i, $i = R, S, D, P, T, Y$, are the total flowrates of the various streams. The species F_A and F_B are mixed with the recycle stream F_T that enters a continuously stirred tank reactor (CSTR), where the following three reactions take place:

$$A + B \xrightarrow{\mathsf{k}_1} C, \tag{4.27a}$$
$$C + B \xrightarrow{\mathsf{k}_2} P + E, \tag{4.27b}$$
$$P + C \xrightarrow{\mathsf{k}_3} G. \tag{4.27c}$$
$$\mathsf{k}_i = \mathsf{A}_i \exp\left(\frac{-\mathsf{E}_i}{T_r + 273.15}\right); \quad i = 1, 2, 3. \tag{4.27d}$$

Here, C is an intermediate, P is the main product, E is a side product, and G is an oily waste product. The side product E can be sold for its fuel value, while G must be disposed off at a cost. The decanter completely separates Species G from the reactor outlet

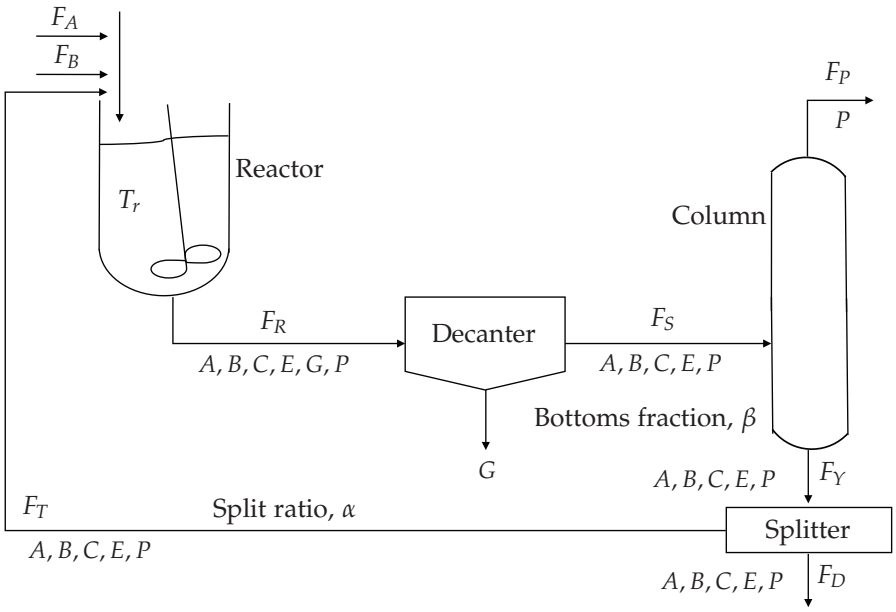


Figure 4.1 – Willliams-Otto process

stream F_R into the waste product stream F_G. The decanter outlet stream F_S is sent to a distillation unit that separates the product P. Due to the formation of an azeotrope, some of the product P (in fact, the fraction β of the mass flowrate of component E in the feed) is retained in the column bottoms. Most of this bottom product is recycled to the reactor, while the rest is used as fuel. The reactor is simplified by assuming isothermal operation. The other units are also simplified to keep the example small and illustrate the proposed concepts with lesser complexity. As a result, the process is modeled without an energy balance. The material balance equations are described in Appendix B.1. The optimization objective is to maximize the return on investment (ROI) given in terms of the net sales minus fixed charges, raw material, utility, and waste disposal costs. It is also desired to keep the production of the product P below a certain threshold. The optimization problem can be formulated mathematically as follows:

$$\max_{F_A, F_B, T_r, \alpha} \text{ROI} := 7358.4\,(P_P\,F_P + P_D\,F_D) - 8400\,(P_A\,F_A + P_B\,F_B + P_G\,F_G) - P_R\,F_R, \quad (4.28a)$$

$$\text{s.t.} \quad \text{(B.1)} - \text{(B.4)}. \quad (4.28b)$$

$$F_P\ (\text{kg/s}) \leq 0.7, \quad (4.28c)$$

$$1 \leq F_A\ (\text{kg/s}) \leq 5, \quad (4.28d)$$

$$1 \leq F_B\ (\text{kg/s}) \leq 4.5, \quad (4.28e)$$

$$70 \leq T_r\ (^\circ\text{C}) \leq 100, \quad (4.28f)$$

$$0 \leq \alpha \leq 0.95. \quad (4.28g)$$

The price values P_i, $i = A, B, D, G, P, R$, are given in Table B.1. The optimal inputs are $\mathbf{u}_p^\star = [1.79, 4.04, 85.82, 0.89]^\mathsf{T}$ with the ROI value of 725.12.

Alternatively, the CSTR can be modeled by two reactions, thereby ignoring the intermediate C:

$$A + 2B \xrightarrow{\bar{\mathsf{k}}_1} P + E, \tag{4.29a}$$

$$A + B + P \xrightarrow{\bar{\mathsf{k}}_2} G. \tag{4.29b}$$

$$\bar{\mathsf{k}}_i = \bar{\mathsf{A}}_i \exp\left(\frac{-\bar{\mathsf{E}}_i}{T_r + 273.15}\right); \quad i = 1, 2. \tag{4.29c}$$

The corresponding model equations are given in (B.5). The corresponding model equations are given in (B.5). The optimal input value with this model are $\mathbf{u}^\star = [1.08, 2.02, 100, 0.67]^\mathsf{T}$. Note that this value is very different then the corresponding value $\mathbf{u}_p^\star$ for the perviously described model.

The proposed RTO scheme is tested in simulation, with the three-reaction Model (B.1) serving as plant substitute. This simulated plant is treated as a black-box whose output measurements are available, thus resembling a real process system. We assume that the plant optimum $\mathbf{u}_p^\star$ is unknown. Consequently, Model (B.5) is adapted iteratively to determine the unknown plant optimum. The adjustable model parameters are the pre-exponential factors $\bar{\mathsf{A}}_1$ and $\bar{\mathsf{A}}_2$, the activation energies $\bar{\mathsf{E}}_1$ and $\bar{\mathsf{E}}_2$, and the bottoms fraction β. The nominal values of the parameters and their uncertainty range are given in Table B.2. The simulations are carried out assuming that additive Gaussian noise acts on the cost and constraint measurements. The standard deviations of the noise on the cost and constraints are 0.5 and 0.005, respectively. The RTO scheme is initialized at $\mathbf{u}_0 = [1, 4.2, 82.57, 0.3]^\mathsf{T}$. The first step consists of optimizing the model at the nominal parameter values $\boldsymbol{\theta}_0$ to compute $\mathbf{u}_1^\star$ ($= \mathbf{u}^\star$). Then, the inputs are updated according to (4.14) to give $\mathbf{u}_1 = [1.015, 3.98, 86.05, 0.45]^\mathsf{T}$. The structural properties of the model at $\mathbf{u}_1$ are analyzed next to suggest the proper KKT elements to parameters pairing for parameter estimation.

4.4.2 Model Structural Independence

There are $n_\mathrm{u} + n_\mathrm{g} = 5$ KKT elements, and the total number of parameters is also $n_\theta = 5$. Hence, ideally, each KKT element should be paired with a single direction in parameter space so that each KKT element can be matched independently of the other elements as per (4.5).

At $\mathbf{u}_1$, the influential parameter space for the i^{th} KKT element $\mathcal{M}_{i,1}$ is found by computing the corresponding matrix $\boldsymbol{C}_1^{(i)}$ via Algorithm 2.1 given in Appendix B. The eigenvalues of $\boldsymbol{C}_1^{(i)}$ are plotted in Figure 4.2 for $i = 1, \ldots, (n_\mathrm{u} + n_\mathrm{g})$. A large gap be-

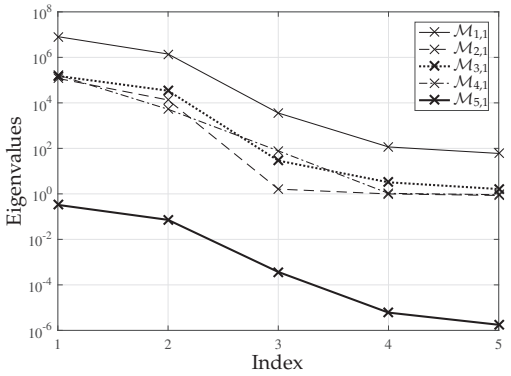


Figure 4.2 – Eigenvalue plots for different KKT elements

tween the second and the third eigenvalue implies that a two-dimensional influential parameter space exists for that KKT element. In fact it turns out that each of the 5 KKT elements have a two-dimensional influential parameter space.

To verify whether there exists at least one parameter direction that is unique to each KKT element, the components of the first two eigenvectors of $\boldsymbol{C}_1^{(i)}$ are plotted. Figure 4.3a shows that, for each KKT element, the absolute value of the eigenvector components corresponding to the two activation energy parameters $\bar{\mathsf{E}}_1$ and $\bar{\mathsf{E}}_2$ dominate, whereas the components of the eigenvector corresponding to the other three parameters are negligible. Figure 4.3b shows that a similar dominance of $\bar{\mathsf{E}}_1$ and $\bar{\mathsf{E}}_2$ is found in the second eigenvector. This implies that each KKT element can only be matched by adjusting either $\bar{\mathsf{E}}_1$ or $\bar{\mathsf{E}}_2$ or a linear combination of the two parameters. Since both parameters influence all the KKT elements, we can only pair one KKT element with one parameter direction. The remaining KKT elements cannot be matched independently by adjusting other parameters along the remaining orthogonal directions. To choose the best KKT element for pairing, it is proposed to determine the privileged input directions, as discussed next.

4.4.3 Privileged Directions and Corresponding Influential Spaces

To discover privileged directions, the global sensitivity analysis approach via Algorithm 3.2 is applied. Figure 4.4a shows the eigenvalues of the matrix $\hat{\boldsymbol{A}}_k$ computed at $\mathbf{u}_1$. Since there is a large gap between the first and the second eigenvalues, the first eigenvector gives the direction that is highly sensitive to parametric variations as compared to the other directions in the input space. Hence, only this eigenvector is considered as a privileged direction. Figure 4.4b shows the components of the first singular vector. The first component has a magnitude close to 1 and indicates that $\mathcal{M}_{1,1}$, namely, the component of the Lagrangian gradient corresponding to the feed rate F_A, is most sensitive to parametric perturbations.

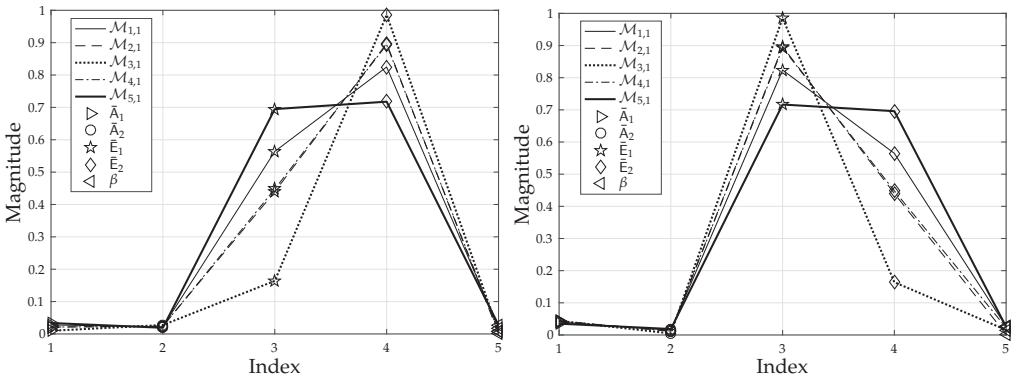


Figure 4.3 – Influential parameter spaces for the various KKT elements at $\mathbf{u}_1$. (a) Absolute value of the components of the first eigenvector. (b) Absolute value of the components of the second eigenvector.

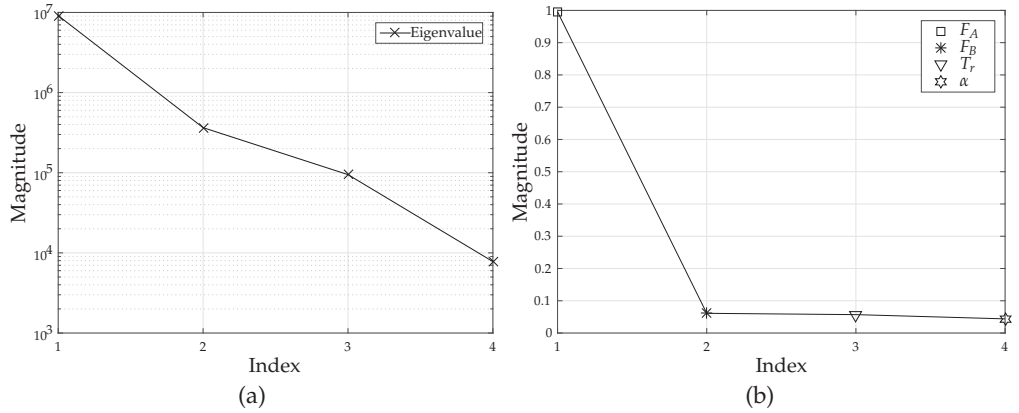
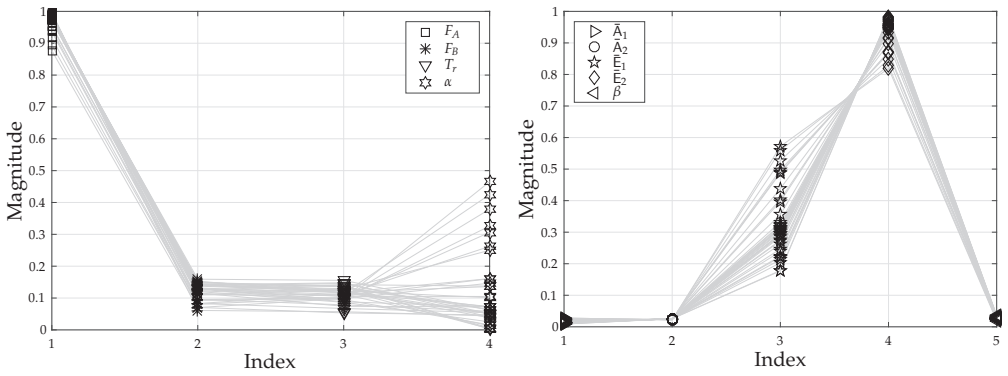


Figure 4.4 – Finding privileged directions at $\mathbf{u}_1$ via Algorithm 3.2. (a) Eigenvalues plot. (b) Absolute value of the components of the first eigenvector.

4.4.4 Matching only the Privileged KKT Elements

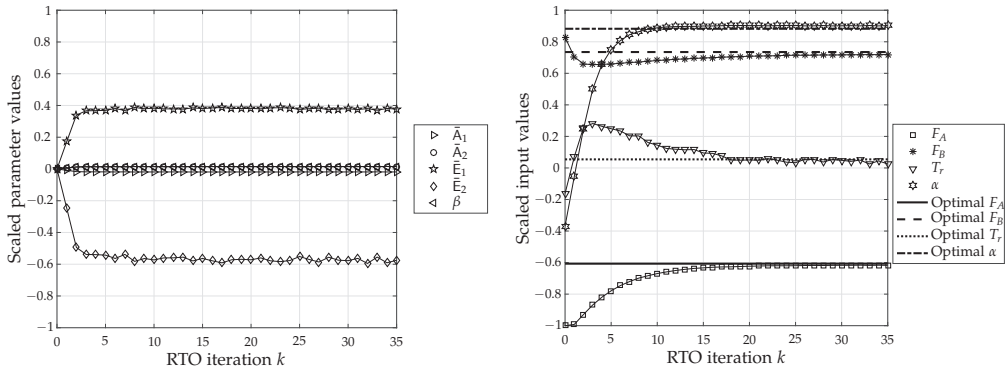
We only consider two KKT elements, $\mathcal{M}_{5,1}$ corresponding to the constraint function and the directional derivative along the privileged direction. Since the KKT element $\mathcal{M}_{5,1}$ cannot be matched independently by adapting the model parameters, it is paired with the zeroth-order modifier. The RTO scheme proposed for Example 4.4 is used for meeting the NCOs in this simulation study.

The privileged direction is recomputed at each RTO iteration. Figure 4.5a indicates that the KKT element $\mathcal{M}_{1,k}$ dominates at each iteration k. From Figure 4.5b, the first eigenvector of the active subspace corresponding to the privileged direction is mainly a linear combination of the parameters $\bar{\mathsf{E}}_1$ and $\bar{\mathsf{E}}_2$. This linear combination is used to form $\boldsymbol{\theta}^{\text{priv}}$ that is used to match the directional derivatives of the model and the plant



(a) (b)

Figure 4.5 – Pairing privileged direction with its active subspace over RTO iterations. (a) Absolute value of the components of the privileged directions. (b) Absolute value of the components of the first eigenvector of the influential space corresponding to the privileged direction.



(a) (b)

Figure 4.6 – Parameter and input evolution over RTO iterations.

along the privileged input direction. The proposed method leads to the optimal performance seen in Figure 4.7. The evolution of the model parameters and the inputs are plotted in Figure 4.6. Since the privileged directions and the corresponding influential parameter spaces are dependent on the inputs $\mathbf{u}$, the proposed scheme is tested using 10 randomly chosen initial inputs $\mathbf{u}_0$ and each time with different noise realizations on the plant cost and constraint values. The result of the different simulation runs are plotted in Figure 4.8, which shows that all runs successfully reduce the sub-optimality gap to a small value.

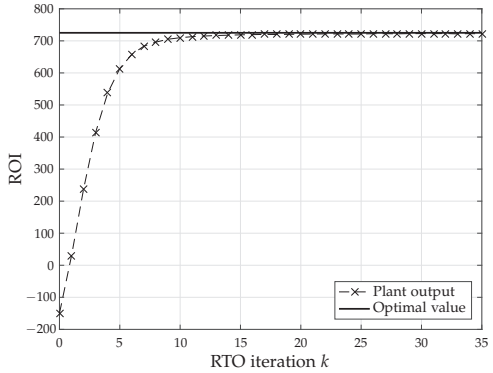


Figure 4.7 – ROI evolution over RTO iterations

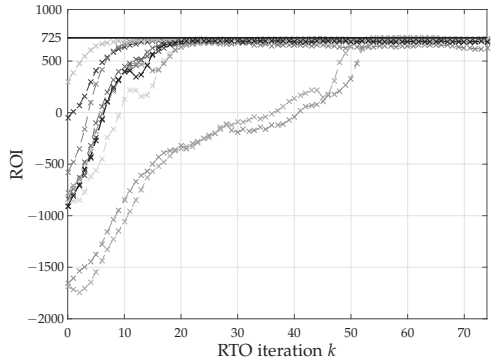


Figure 4.8 – ROI evolution over RTO iterations for different input initialization.

4.5 Summary

The ability of modifier adaptation to reach plant KKT point is attributed to its structural independence property. Structural independence enables one-to-one correspondence between the modifiers and the KKT elements that provides the modified model (2.16) enough flexibility to match all the KKT conditions.

However, if the parameters of a given model provides this flexibility then such parameters can be adapted instead of modifiers. Therefore, such parameters are called as mirror parameters. It is proposed to find the mirror parameters in a given model by finding influential and non-influential parameter spaces of each of the KKT elements. If a KKT element has a parameter direction that lies in the influential parameter space of that KKT element and in the non-influential space of the rest of the KKT elements, then the KKT element and the parameter direction can be paired. If such parameter directions exist for each of the KKT elements then the model is said to be structurally independent. Therefore, parameters can be adapted instead of modifiers for KKT matching. On the other hand if some KKT elements lack such parameter directions than they can be matched by adapting modifiers.

In the scenarios where the models are not structurally independent and the experimental cost of gradient estimation is considered too high, the gradients are estimated only along privileged directions. Similar to KKT element-parameter direction pairing, the privileged directions are paired with corresponding parameter directions. In the case of only partial KKT matching, it is proposed to match the KKT elements corresponding to the constraints by adapting zero-order modifiers when the constraints and the privileged directions compete for a parameter direction. As the experimental cost of gradient matching is higher, the parameter direction is adapted to match the gradient along the privileged directions (instead of adapting that parameter direction to match the constraints).

Finally, a simulation study of Williams-Otto is used to illustrate the presented concepts. A structural plant-model mismatch is introduced by considering only a two reaction system for the model as opposed to a three reaction system for the plant. There are four inputs and one constraint in the optimization problem resulting in a total of five KKT elements. The model consists of five parameters and has five KKT elements to match. Ideally, one should be able to pair each KKT element with one parameter direction in a five-dimensional parameter space. The structural independence analysis reveals that each KKT element is influenced by only two parameter directions that are dominated by mainly two parameters corresponding to the activation energies. Therefore, only one KKT element can be independently paired with only one parameter direction and the constraint is paired with a zero-order modifier.

It is further discovered that the mapping of the model Lagrangian to inputs is most sensitive with respect to the input component corresponding to the feedrate of reactant A. The only privileged direction found had the highest weight corresponding to the feederate of A. Hence, the model Lagrangian gradient is corrected mainly along the input on feedrate of A by adjusting the most influential linear combination of the two activation energies. Despite the structural plant-model mismatch, the proposed RTO scheme converges to near-optimal solutions for every randomly chosen parameter initialization and in the presence of additive noise on the plant cost and constraint measurements.

5 Advanced Case Studies

The concept of measuring only the privileged KKTs is put to test on two different simulation studies. The first study concerns power generation via flying kites. Large flying kites are able to generate power as they convert the mechanical energy imparted by flowing winds to electricity. In the presence of disturbances, such as changing wind conditions and model uncertainties, it becomes important that the path followed by a flying kite is adapted for optimal power generation. Past studies [31, 32, 34] it have shown that MA-based RTO techniques offer a promising solution to the optimal path-following problem of kites. To advance the MA-based optimal control of kites, ADMA is tested on a simulated kite system.

The kite dynamics is highly nonlinear and therefore, under the presence of a plant-model mismatch, very sensitive to input changes [31]. Hence, controlling the input step lengths via static gain filters $\boldsymbol{K}$, applied either to modifier terms or to the inputs directly, is virtually impractical. Trust-region approach [26] offers a solution to such problems by dynamically adapting its size that aims at limiting the plant-model mismatch due to uncorrected second- and higher-order terms [17]. Here, the ADMA approach is further developed to ensure kite stability by incorporating a trust region.

The second case study consists of a fuel-cell system that can be seen as a cogeneration or a combined heat and power generation unit. The system consists of a solid oxide fuel-cell (SOFC) stack that intakes a fuel (typically hydrogen) and an oxidant to generate electricity. The main by-product is water, thus making the SOFC technology a promising renewable source of energy. Since SOFCs typically run continuously for long hours and are subject to changes in the power demand, it is desirable to keep the performance optimal throughout, while ensuring that the operation remains within safety and operability constraints [60, 136]. Due to changes in the power demand during operation, but also due to external perturbations affecting the SOFC system, the set of optimal operating conditions continuously vary with time. In addition, the fuel cell is operated under a number of inequality constraints including bounds on input

and output variables. Constraints on the cell voltage and fuel utilization are set due to risks of oxidation of the cell's anode, which may degrade or even cause the failure of the cell [23, 131]. Operating at high current densities can cause material damage to the cell through excessive heating [76]. Therefore, fuel-cell systems require technologies that can guarantee constraint satisfaction. Hence, there is a need for real-time optimization, i.e., repeated adjustment of the operating variables (for example, flowrates of fuel and air) to maximize the performance (e.g., power output, efficiency) of the fuel-cell system, while satisfying the process constraints.

Past studies have shown that RTO is a suitable technique for a fuel-cell system [20, 82]. Specifically, the RTO technique constraint adaptation (CA) has shown great promise for proper functioning and enhanced life of the fuel-cell system [20, 100]. Although, CA guarantees constraint satisfaction upon convergence, it may well converge to a sub-optimal solution. In such cases, gradient corrections become essential to ensure optimality [21]. Thus, ADMA offers an appealing alternative as it can compensate for sub-optimality at reasonable experimental costs. The second section of the chapter focuses on application of different variants of MA to the fuel-cell system and thereby, discusses various aspects of the applied methods.

5.1 A Simulated Kite System

A dynamically flying kite is a fast, unstable system influenced by unpredictable wind disturbances. The aim of a kite system is to generate power by exploiting the airborne wind energy. To this end, intelligent path planning is important because, although the kite is free to follow any path, it is the flight path that directly determines the aerodynamic force the kite experiences and, hence, the power generated. Experimental studies [134] have confirmed that the path taken by the kite significantly influences the power it can generate. Therefore, this study focuses on finding optimal path for the kite to follow in order to maximze the power generated in the presence of uncertainties.

5.1.1 Kite Dynamics

This simulation study uses the Erhard model [41] that is previously studied in [33] for MA applications. In the Erhard model, the kite is modeled as a point in a spherical coordinate system. Using a fixed tether length, the states are the spherical angles ϑ and φ and the turning angle ψ, the latter being the kite orientation (see [33] for more details). The manipulated variable for controlling the kite is the steering deflection δ. The kite dynamics (as detailed in [31, 33]) are described by the following differential

equations:

$$\begin{aligned} \dot{\vartheta} &= \frac{w_{app}}{r}(\cos\psi - \frac{\tan\vartheta}{E}), \\ \dot{\varphi} &= -\frac{w_{app}}{r\sin\vartheta}\sin\psi, \\ \dot{\psi} &= w_{app}(g_s\delta - \frac{\sin\psi}{r\tan\vartheta}), \\ w_{app} &= wE\cos\vartheta, \end{aligned} \tag{5.1}$$

where w_{app} is the apparent wind speed, r is the fixed kite radius, and w is the wind speed. The other parameters and their values are given in Table 5.1, which also highlights the parametric mismatch between the plant and the model. The plant parameter values match closely the prototypes under development [53, 108]. The glide ratio (lift/drag ratio) E is given by

$$E = E_0 - c\delta^2, \tag{5.2}$$

where c is the turning penalty factor. The wind speed is a function of the altitude z of the kite. To introduce structural plant-model mismatch in addition to parametric mismatch, the following wind speed laws are used

$$w = \begin{cases} w_{ref} + (z - z_{ref})\Delta w & \text{for the model,} \\ w_{ref}(\frac{z}{z_{ref}})^a & \text{for the plant,} \end{cases} \tag{5.3}$$

with $z = r\sin\vartheta\cos\varphi$, and where a is the surface friction coefficient and w_{ref} the reference wind speed at the reference altitude z_{ref}. The wind-altitude relationship is linear for the model and follows a power law for the plant [5].

The objective is to maximize the average line tension T over one loop,

$$T = \int_0^{tf} (\frac{1}{2}\rho A w^2)(E+1)\sqrt{E^2+1}\cos^2\vartheta \, dt, \tag{5.4}$$

where t_f is the time period of one loop. The standard constraints for kites are the periodicity of the path, a minimal altitude, bounds on the states ϑ and φ, and the saturation of the input δ:

$$\begin{aligned} &\text{altitude constraint } z = r\sin\vartheta\cos\varphi \geq z_{min}, \\ &\text{bounds} \begin{cases} 0 \leq \vartheta(t) < \frac{\pi}{2}, \\ -\frac{\pi}{2} \leq \varphi(t) \leq \frac{\pi}{2}, \\ -\delta_{max} \leq \delta(t) \leq \delta_{max}, \end{cases} \end{aligned} \tag{5.5}$$

where t is the time. Note that there is no bound on the third state ψ.

Table 5.1 – Plant and model parameters. The uncertain model parameters $\boldsymbol{\theta}$ are highlighted in blue.

Parameter	Description	Plant value	Nominal model value	Uncertainty interval (uniform distribution)
r (m)	tether length	250	250	-
A (m^2)	surface of the kite	25	25	-
ρ (kg/m^3)	air density	1.2	1.2	-
E_0	initial lift to drag ratio	6	4.5	$\pm$11%
g_s (rad/m^2)	turning constant factor	$5 \cdot 10^{-3}$	$7 \cdot 10^{-3}$	$\pm$14%
c (1/m^2)	turning penalty factor	0.06	0.02	$\pm$10%
z_{ref} (m)	reference altitude	10	10	-
w_{ref} (m/s)	reference wind speed	8	8	-
a	surface friction coefficient	0.15	-	-
Δw (1/s)	wind speed change rate	-	10^{-3}	$\pm$5%
δ_{max} (m)	max of steering deflection	7.5	7.5	-
z_{min} (m)	minimal altitude	12.5	12.5	-

5.1.2 Reference Trajectory Parametrization

The optimization layer is used to generate an optimal reference trajectory for the kite such that the line tension is maximized. As the kites used for power generation are unstable, a controller must continuously adjust the steering deflection δ to ensure that the kite does not crash. In this simulation study, it is assumed that a perfect path-following controller exists, which ensures that the kite follows a given periodic reference path.

For optimization, the reference trajectory is parametrized as a smooth curve. Thus, the decision variables for the optimization are the curve parameters. Such a formulation allows to use MA as a run-to-run optimization scheme. Here, the parametrization of the reference is a closed Bézier curve defined as a polynomial for $\alpha \in [0, 1]$.

$$\mathbf{F}(\alpha) := \sum_{i=0}^{n} \binom{n}{i} (1-\alpha)^{i-1} \alpha^i \mathbf{P}_i, \tag{5.6}$$

where $\mathbf{P}_i \in \mathbb{R}^2$ are the curve parameters. The polynomial is used to represent the (ϑ, φ) references with $\mathbf{F}(\alpha) = [\vartheta(\alpha), \varphi(\alpha)]^T$. The dynamics (5.1) is scaled with respect to time to make $\alpha = 1$ represent the full time period t_f using $\dot{\alpha} = 1/t_f$. The objective (5.4) is maximized with respect to this scaled dynamics using TR-ADMA with the decision variables $\mathbf{P}_i, i = 0 \dots n$, and t_f, with $n = 7$. Hence, the optimal curve parameters $\mathbf{P}_i^\star$ lead to a dynamically feasible periodic optimal trajectory. The resulting smooth parameterized curve can then be tracked as done in [39]. Based on the approximation of the Erhard model in crosswind flight, one can show that the apparent wind speed w_{app} can be approximated by the kite speed w_k [34]. ψ and δ are represented in terms

Table 5.2 – Investigated cases (NA: not applicable)

Case	1	2	3
$n_{\max}$	2	2	2
$N(samples)$	500	500	500
trust region	**no**	**yes**	**yes**
noise	**0**	**0**	**3**
Filter eigenvalues $\boldsymbol{K}^{\varepsilon^{\Phi}}$,$\boldsymbol{K}^{\varepsilon}$	0.1	-	0.95
Filter eigenvalues $\boldsymbol{K}^{\mathrm{G}_i}$, $\boldsymbol{K}^{\Phi}$	0.1	-	0.25
optimality loss	NA	2%	$7.1 \pm 2.2\%$

of the (ϑ, φ)-curve by approximating them as

$$\psi \approx -\gamma = -\tan^{-1}\left(\frac{\dot{\varphi}\cos\vartheta}{\dot{\vartheta}}\right), \text{ and } \delta \approx -\frac{\dot{\gamma}}{g_s w_k} \tag{5.7}$$

5.1.3 Initial Results and the Need for Step-Length Control

To apply ADMA, the number of privileged directions is fixed. Choosing the number of privileged directions is a trade-off between the number of plant experiments for gradient/directional derivative estimation and the perceived optimality loss. However, one can always start with a reasonably low number of privileged directions. If required, then upon convergence, more plant experiments can be performed to estimate the gradient in the remaining directions. If the gradient error is high in the remaining directions, then appropriate model corrections can be made to further optimize the underlying system.

Here, the number of privileged directions is fixed at $n_{\max} = 2$. For computing the privileged direction matrix $\hat{\boldsymbol{W}}_k$, a parametric uncertainty interval of $\pm(5 \text{ to } 14)\,\%$ around the respective nominal model values (given in Table 5.1) is considered. Monte-Carlo samples are collected within the uncertainty interval by assuming a uniform probability distribution. To estimate the directional derivatives, forward finite-difference experiments are performed on the plant along the privileged directions.

Initially, ADMA Algorithm 3.3 is applied to the kite system without considering any noise in the plant outputs and with the classic modifier filters applied in ADMA. This scenario is referred to as Case 1 and the parameters utilized in Algorithm 3.3 for Case 1 are given in Table 5.2. The obtained results are plotted in Figure 5.1. The figure shows that ADMA drives the plant towards its optimum but then the system becomes unstable and starts oscillating. This is due to the fact that the model Hessian

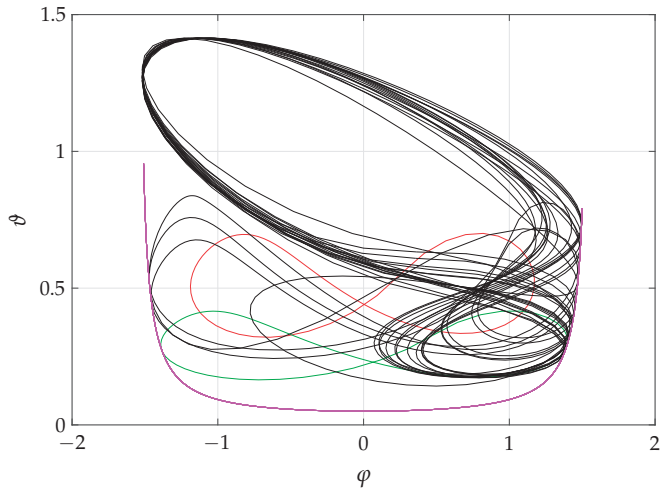


Figure 5.1 – Case 1. ADMA Algorithm 3.3 applied to the simulated kite. Green - Model optimal path, Red - Plant optimal path, Black - Plant behavior over RTO iterations, Magenta - Altitude constraint.

is non-convex at and around the plant optimum and the plant's Hessian is obviously convex at the plant optimum. Since, by construction, only the first-order corrections are enforced and the second-order (Hessian) corrections are not made, ADMA *jumps* ahead of the plant optimum and needs to start again far away from it.

This issue can be overcome by adjusting the model Hessian. However, correcting the model Hessian by plant measurements is even more experimentally intense. Moreover, even if sufficient plant data is gathered for Hessian estimation, the numerical errors and the presence of noise can pose significant challenges in accurate Hessian estimation. An alternative is to use trust region and directly control the input step lengths. This way, the Hessian corrections can be avoided without compromising on optimality. Moreover, [17] showed how MA is equivalent to trust-region-based optimization framework. In [17] and [12], it is further shown that controlling the input step lengths in MA via trust region is a natural choice for guaranteeing convergence. Thus, ADMA framework is developed further to incorporate trust-region to control the change in the input $\mathbf{u}$ from one RTO iteration to the next.

5.1.4 Trust Region for Input Step Length Control

The input $\mathbf{u}_{k+1}$ obtained in ADMA Algorithm 3.3 may be far away from the previous input $\mathbf{u}_k$. As the corrections applied to the cost and constraint functions are only locally valid around the operating point determined at the k^{th} iteration, the application of $\mathbf{u}_{k+1}$ to the plant may result in poor performance. To limit the over-reliance on the model, a trust region around the current inputs is defined, wherein the modified model is a fairly accurate representation of the plant. The modified-optimization prob-

lem of ADMA is then solved inside the trust region so as to iteratively improve the plant performance. The size of the trust region is critical and is determined based on the predicted (model-based) and actual (plant-based) performance. The use of a trust region in MA was shown to result in global convergence [17]. Given the benefits of the trust-region concept, it is proposed to include a trust region within ADMA, thereby resulting in the TR-ADMA algorithm described as Algorithm 5.1.

Trust-Region Adjustment in TR-ADMA

The key element in any trust-region-based algorithm is the choice of its size $\Delta_k \in \mathbb{R}^{n_u}$ at each iteration. This choice is based on the agreement between the performance predicted by the model and the actual plant performance. To this end, the following ratio between the plant Lagrangian and the Lagrangian of the modified model is introduced

$$\varrho_k = \frac{\mathcal{L}_p(\mathbf{u}_k, \boldsymbol{\mu}_k) - \mathcal{L}_p(\mathbf{u}^\star_{k+1}, \boldsymbol{\mu}^\star_{k+1})}{\mathcal{L}_{m,k}(\mathbf{u}_k, \boldsymbol{\mu}_k) - \mathcal{L}_{m,k}(\mathbf{u}^\star_{k+1}, \boldsymbol{\mu}^\star_{k+1})}, \tag{5.9}$$

where $\mathcal{L}_p(\mathbf{u}, \boldsymbol{\mu}) := \Phi_p(\mathbf{u}) + \boldsymbol{\mu}^\mathsf{T}\mathbf{G}_p(\mathbf{u})$ is the plant Lagrangian and $\mathcal{L}_{m,k}(\mathbf{u}, \boldsymbol{\mu}) := \Phi_{m,k}(\mathbf{u}) + \boldsymbol{\mu}^\mathsf{T}\mathbf{G}_{m,k}(\mathbf{u})$ is the modified-model Lagrangian, with $\mathbf{G}_p$ denoting the vector of plant constraints $\mathrm{G}_{p,i}$ and $\mathbf{G}_{m,k}$ denoting the vector of model constraints $\mathrm{G}_{m,i,k}$. Based on the value of the ration ϱ_k, it is decided whether to enlarge the trust region by a factor $v_1 > 0$, keep it constant or decrease the trust region by a factor of $v_2 > 0$.

At k^{th} iteration, the solution to Problem (5.8) in Algorithm 5.1 is $(\mathbf{u}^\star_{k+1}, \boldsymbol{\mu}^\star_{k+1})$. The ratio ϱ_k is evaluated at $(\mathbf{u}^\star_{k+1}, \boldsymbol{\mu}^\star_{k+1})$. Decision regarding accepting the solution $(\mathbf{u}^\star_{k+1}, \boldsymbol{\mu}^\star_{k+1})$ is also taken based on the value of the ratio ϱ_k. To this end, the scalar parameters v, γ_1 and γ_2 are defined with $0 \leq v \leq \gamma_1 < \gamma_2 < 1$. Then, the trust region is adapted as follows

- $\varrho_k < 0$ implies that the modified model predicts a decrease in Lagrangian value, while the plant Lagrangian value actually increases, or vice-versa. The model gives a wrong prediction as it is not sufficiently accurate in the prevailing trust region. Hence, the input vector $\mathbf{u}^\star_{k+1}$ is rejected, and the trust region is reduced in order to find a region in which the corrected model can be trusted.
- For $0 \leq \varrho_k < v$, the modified model predicts a large change, while the plant differs only a little from the previous iteration, thus indicating a large disagreement between the two. Hence, the trust region is decreased.
- For $v \leq \varrho_k \leq \gamma_1$, the changes in Lagrangians are sufficiently similar to accept the new input vector $\mathbf{u}^\star_{k+1}$, but the trust region is still decreased as the model prediction is not sufficiently close to the actual performance.
- For $\gamma_1 \leq \varrho_k < \gamma_2$, the prediction is good as it mostly agrees with the actual

Algorithm 5.1 TR-ADMA

Step 0 (Initialization): Choose the eigenvalues of the (typically diagonal) filter matrices $\boldsymbol{K}^{\varepsilon^{\Phi}}$, $\boldsymbol{K}^{\Phi}$, $\boldsymbol{K}^{\varepsilon}$ and $\boldsymbol{K}^{\mathrm{G}_i}$ in the interval]0,1]. Initialize the input vector $\mathbf{u}_0$. Set the modifiers $\varepsilon_0^{\Phi} = 0$, $\boldsymbol{\lambda}_0^{\Phi} = \mathbf{0}$, $\varepsilon_0^{\mathrm{G}_i} = 0$ and $\boldsymbol{\lambda}_0^{\mathrm{G}_i} = \mathbf{0}$. Choose the maximal allowable step size Δ^{max}, the parameters $0 < v_1 < 1$, $v_2 > 1$, the updating range parameters $0 \leq v \leq \gamma_1 < \gamma_2 < 1$ and set Δ_0 to an arbitrarily large value.

for $k = 0 \rightarrow \infty$

Step 1. Compute

$$(\mathbf{u}_{k+1}^{\star}, \boldsymbol{\mu}_{k+1}^{\star}) = \underset{\mathbf{u}}{\operatorname{argmin}}\ \Phi_{m,k}(\mathbf{u}) \tag{5.8a}$$

$$\text{s.t. } \mathrm{G}_{m,i,k}(\mathbf{u}) \leq 0,\ i = 1, \ldots, n_{\mathrm{g}} \text{ and } \mathbf{u}_k - \Delta_k \leq \mathbf{u} \leq \mathbf{u}_k + \Delta_k. \tag{5.8b}$$

Step 2. Apply $\mathbf{u}_{k+1}^{\star}$ to the plant and measure the noisy cost $\widetilde{\Phi}_p(\mathbf{u}_{k+1}^{\star})$ and noisy constraints $\widetilde{\mathbf{G}}_p(\mathbf{u}_{k+1}^{\star})$ and thereby compute, $\widetilde{\mathcal{L}}_p(\mathbf{u}_{k+1}^{\star}, \boldsymbol{\mu}_{k+1}^{\star})$.

Step 3. Update the step size

if $k = 0$, then $\varrho_k = 1$

else $\varrho_k = \frac{\widetilde{\mathcal{L}}_p(\mathbf{u}_k, \boldsymbol{\mu}_k) - \widetilde{\mathcal{L}}_p(\mathbf{u}_{k+1}^{\star}, \boldsymbol{\mu}_{k+1}^{\star})}{\mathcal{L}_{m,k}(\mathbf{u}_k, \boldsymbol{\mu}_k) - \mathcal{L}_{m,k}(\mathbf{u}_{k+1}^{\star}, \boldsymbol{\mu}_{k+1}^{\star})}$

endif

if $\varrho_k > \gamma_2$, then $\Delta_{k+1} = \min\,(\ v_2 \cdot \Delta_k,\ \Delta^{max}\ )$

elseif $\varrho_k < \gamma_1$, then $\Delta_{k+1} = v_1 \cdot \Delta_k$

else $\Delta_{k+1} = \Delta_k$

endif

if $\varrho_k \geq v$, then $(\mathbf{u}_{k+1}, \boldsymbol{\mu}_{k+1}) := (\mathbf{u}_{k+1}^{\star}, \boldsymbol{\mu}_{k+1}^{\star})$.

else $(\mathbf{u}_{k+1}, \boldsymbol{\mu}_{k+1}) = (\mathbf{u}_k, \boldsymbol{\mu}_k)$, $\hat{\boldsymbol{W}}_{1,k+1} = \hat{\boldsymbol{W}}_{1,k}$, $(\varepsilon_{k+1}^{\Phi}, \boldsymbol{\varepsilon}_{k+1}^{\mathrm{G}}, \boldsymbol{\lambda}_{k+1}^{\Phi}, \boldsymbol{\lambda}_{k+1}^{\mathrm{G}_i}) = (\varepsilon_k^{\Phi}, \boldsymbol{\varepsilon}_k^{\mathrm{G}}, \boldsymbol{\lambda}_k^{\Phi}, \boldsymbol{\lambda}_k^{\mathrm{G}_i})$ and $(\widehat{\nabla\Phi}_{k+1}, \widehat{\nabla\mathrm{G}}_{i,k+1}) = (\widehat{\nabla\Phi}_k, \widehat{\nabla\mathrm{G}}_{i,k})$ for $i = 1, \ldots, n_{\mathrm{g}}$ and goto step 1.

endif

Step 4. Find the privileged directions matrix $\hat{\boldsymbol{W}}_{1,k+1}$ using Algorithm 3.2 and estimate the directional derivatives of the plant cost $\nabla_{\hat{\boldsymbol{W}}_{1,k+1}} \Phi_p(\mathbf{u}_{k+1})$ and constraints $\nabla_{\hat{\boldsymbol{W}}_{1,k+1}} \mathrm{G}_{p,i}(\mathbf{u}_{k+1})$, for example, via finite differences. the full gradients are computed as $\widehat{\nabla_{\mathbf{u}}\Xi_p}(\mathbf{u}_{k+1}) := \nabla_{\mathbf{u}}\Xi(\mathbf{u}_{k+1}, \boldsymbol{\theta}_0)(\boldsymbol{I}_{n_{\mathrm{u}}} - \hat{\boldsymbol{W}}_{1,k+1}\hat{\boldsymbol{W}}_{1,k+1}^{+}) + \nabla_{\hat{\boldsymbol{W}}_{1,k+1}}\Xi_p(\mathbf{u}_{k+1})\hat{\boldsymbol{W}}_{1,k+1}^{+}$.

Step 6. Update the modifiers

$$\begin{aligned}
\varepsilon_{k+1}^{\Phi} &= (\boldsymbol{I}_{n_{\mathrm{g}}} - \boldsymbol{K}^{\varepsilon^{\Phi}})\varepsilon_k^{\Phi} + \boldsymbol{K}^{\varepsilon^{\Phi}}(\widetilde{\Phi}_p(\mathbf{u}_{k+1}) - \Phi(\mathbf{u}_{k+1}, \boldsymbol{\theta})) \\
\boldsymbol{\varepsilon}_{k+1}^{\mathrm{G}} &= (\boldsymbol{I}_{n_{\mathrm{g}}} - \boldsymbol{K}^{\varepsilon})\boldsymbol{\varepsilon}_k^{\mathrm{G}} + \boldsymbol{K}^{\varepsilon}\big(\widetilde{\mathbf{G}}_p(\mathbf{u}_{k+1}) - \mathbf{G}(\mathbf{u}_{k+1}, \boldsymbol{\theta}_0)\big), \\
\boldsymbol{\lambda}_{k+1}^{\Phi} &= (\boldsymbol{I}_{n_{\mathrm{u}}} - \boldsymbol{K}^{\Phi})\boldsymbol{\lambda}_k^{\Phi} + \boldsymbol{K}^{\Phi}\big(\widehat{\nabla_{\mathbf{u}}\Phi_p}(\mathbf{u}_{k+1}) - \nabla_{\mathbf{u}}\Phi(\mathbf{u}_{k+1}, \boldsymbol{\theta}_0)\big)^{\mathsf{T}}, \\
\boldsymbol{\lambda}_{k+1}^{\mathrm{G}_i} &= (\boldsymbol{I}_{n_{\mathrm{u}}} - \boldsymbol{K}^{\mathrm{G}_i})\boldsymbol{\lambda}_k^{\mathrm{G}_i} + \boldsymbol{K}^{\mathrm{G}_i}\big(\widehat{\nabla_{\mathbf{u}}\mathrm{G}_{p,i}}(\mathbf{u}_{k+1}) - \nabla_{\mathbf{u}}\mathrm{G}_i(\mathbf{u}_{k+1}, \boldsymbol{\theta}_0)\big)^{\mathsf{T}}, \quad i = 1, \ldots, n_{\mathrm{g}}.
\end{aligned}$$

endfor

performance. Hence, the input vector $\mathbf{u}_{k+1}^{\star}$ is accepted and the trust region Δ_k is kept at the same value for the next iteration.

- For $\gamma_2 \leq \varrho_k$, the agreement between prediction and actual performance is excellent. The input vector $\mathbf{u}_{k+1}^{\star}$ is accepted and the trust region is enlarged.

Remark 5.1. *Note that the TR-*ADMA *algorithm is a heuristic approach that is used to enforce convergence. However, to guarantee global convergence, any trust-region approach requires to satisfy the Cauchy decrease condition (see [26]) which is not enforced here. Moreover, the local gradient corrections are only partial, thus resulting in gradient errors. Hence, global convergence properties are difficult to achieve. For the locally corrected models, it has been shown in [12] that, if the gradient error is bounded, then global convergence can be guaranteed. It is of future interest to establish such properties for TR-*ADMA*.*

5.1.5 Optimization Results

Here 3 cases are discussed and their important features are summarized in Table 5.2. The trust-region parameters used is Cases 2 and 3 are given in Table 5.3. The measure of performance is the optimality loss, that is, the percentage of improvement upon convergence compared to the plant optimum:

$$\text{optimality loss} := \frac{\Phi_p(\mathbf{u}_p^{\star}) - \Phi_p(\mathbf{u}_{\infty})}{\Phi_p(\mathbf{u}_p^{\star})}, \tag{5.10}$$

with $\mathbf{u}_{\infty}$ being the converged input value.

Recall that Case 1 considers the application of ADMA using noise-free measurements and no trust region. The left plot at the top of the Figure 5.2 compares the predicted and the actual plant behavior in Case 1. The large jump in model prediction is the result of letting the plant free to jump far away from one iteration to the next. Hence, without effective step-length control, the method is not applicable to real cases. Even if full gradient corrections are made as is traditionally done in MA, plant optimality is not reached. Indeed, MA also results in oscillations. A better performance can be achieved by formulating a convex problem at the price of reduced model accuracy. Alternatively, the incorporation of a trust region can help control the step length, thereby enforcing convergence, while avoiding large jumps.

Case 2 considers TR-ADMA using noise-free measurements. As a result, the plant exhibits a perfect behavior as seen in the middle plots of Figure 5.2. Large improve-

Table 5.3 – Trust-region parameters

v	γ_1	γ_2	Δ_{max}	v_1	v_2
0.015	1/4	3/4	0.1	1/2	2

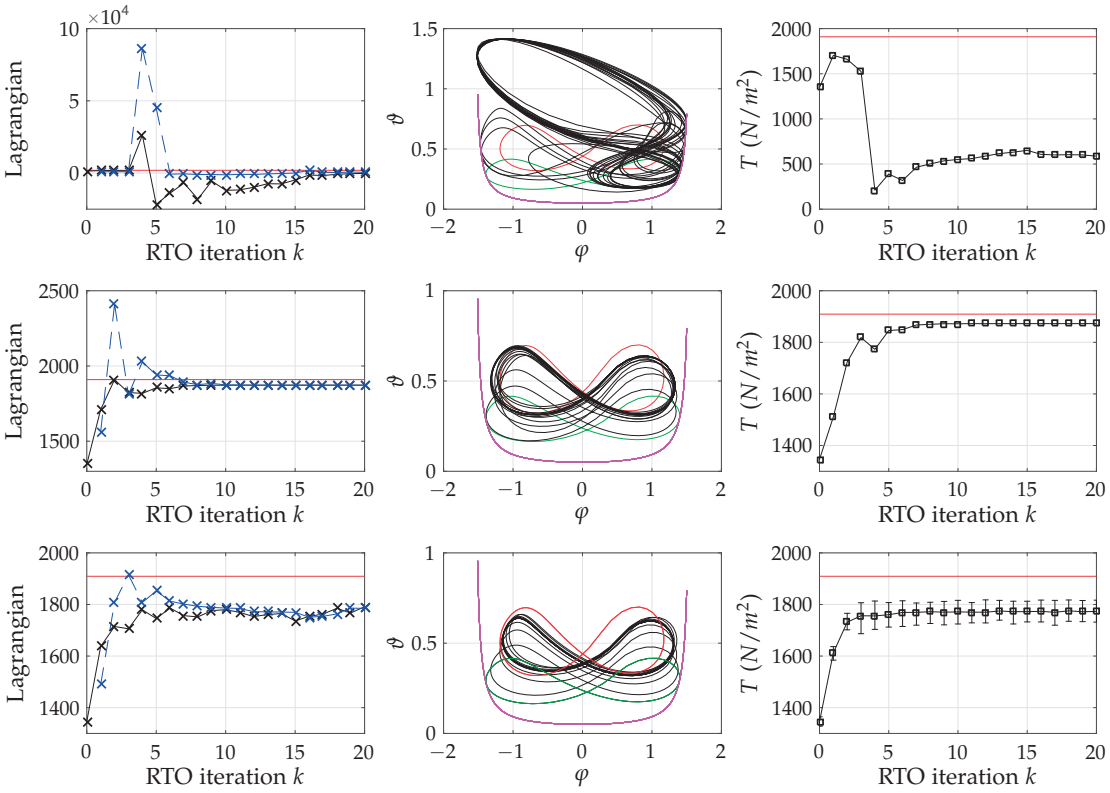


Figure 5.2 – ADMA applied to the simulated kite. **Top plots:** Case 1. **Middle plots:** Case 2. **Bottom plots:** Case 3. The first two plots correspond to a single noise realization, while the last plot of Case 3 shows the average performance over 40 noise realizations. **Left plots**: Black solid lines - Plant Lagrangian, Blue dashed lines - Model Lagrangian, Red - Plant Lagrangian at the plant optimum. **Center Plots**: Green - Model optimal path, Red - Plant optimal path, Black - Plant behavior over RTO iterations, Magenta - Altitude constraint. **Right plots**: Black - Plant output evolution. Red - Optimal line tension.

ment in plant performance happens during the first 7 iterations, after which the algorithm slows down as the trust region adjusts itself to avoid jumping around the plant optimum. The left plot in the middle of the Figure 5.2 shows how the trust-region adjustment decisions are made. A poor prediction causes tightening of the trust region, thereby leading to a better agreement between prediction and actual performance.

As observed in Case 2, the choice of only 2 privileged directions appear to be appropriate in the sense that they lead to little optimality loss. More input directions could be chosen, which however will inevitably result in a larger experimental effort for gradient evaluation. It has also been observed that the incorporation of 4 privileged directions reduces the optimality loss very close to zero (not shown here).

Case 3 considers TR-ADMA and noisy measurements. An additive Gaussian noise with a magnitude of 3 % of the nominal values is added to the plant measurements. The finite-difference scheme gives only rough estimates of the plant directional derivatives. Therefore, the modifiers need filtering. The bottom plots of Figure 5.2 show that it is possible to obtain good performance if the filtering is appropriate. Note that, even with high noise, convergence is achieved. Since the gradient corrections are not so accurate due to the noise, the trust region shrinks, thereby resulting in slower convergence. For Case 3, 40 simulation runs are performed with each run having different random noise realization. The TR-ADMA algorithm converges every time with an average optimality loss of 7.1 %.

5.2 Fuel-Cell System

This fuel-cell system is studied under a joint project that includes the Automatic Control Laboratory of EPFL, group of Energy Materials at EPFL and HTceramix SA-SOLIDpower. The simulation study presented here is part of this project that aims at modelling and optimization of the system. In this section various RTO schemes are tested and compared, including ADMA.

Note that the system described here is a an approximation of the actual system studied. The description of actual system is not provided because of confidentiality reasons.

System Components Description

The system consists of a solid oxide fuel cell with other components that are needed for its proper functioning. These components include a reformer, a heat exchanger system and a burner that are interconnected. The layout of the fuel-cell system is described in Figure 5.3. A small description of different components of the system is as following.

Reformer: The fuel used in the system is the methane gas. It is converted into hydrogen which is finally consumed in the fuel-cell stack to produce electricity. The conversion to hydrogen is done through steam reforming and water-gas shift that takes place in the reformer. It involves the following reactions

$$CH_4 + H_2O \rightarrow CO + 3H_2 \quad \Delta H_r = 206.1\,\text{kJ mol}^{-1}, \tag{5.11}$$

$$CO + H_2O \rightarrow CO_2 + H_2 \quad \Delta H_r = -41.15\,\text{kJ mol}^{-1}. \tag{5.12}$$

The overall reaction system in reforming is endothermic. Therefore, methane entering the reformer is preheated. The preheated water is supplied by the heat exchanger system. The output of the reformer is syngas that is sent to the heat exchanger system

for preheating. Note that the methane to hydrogen conversion is fixed at 99%.

Fuel-cell stack: This is the principle component of the system that allows producing the electrical power. It takes hydrogen present in the syngas and oxidizes it with the hot air supplied by the heat exchanger system. In another reaction, it performs an internal reforming of the residual methane present in the syngas stream:

$$CH_4 + 2H_2O \rightarrow CO_2 + 4H_2, \tag{5.13}$$

$$H_2 + \frac{1}{2}O_2 \rightarrow H_2O. \tag{5.14}$$

Burner: The output gas flow of the fuel cell stack consists of a small amount of methane, hydrogen and carbon monoxide. The gas mixture is sent to the burner that heats them to produce hot gases:

$$CH_4 + 2O_2 \rightarrow CO_2 + 2H_2O, \tag{5.15}$$

$$H_2 + \frac{1}{2}O_2 \rightarrow H_2O, \tag{5.16}$$

$$CO + \frac{1}{2}O_2 \rightarrow CO_2. \tag{5.17}$$

The produced energy is then utilized by the different components of the fuel-cell system.

Heat exchanger system: The heat exchanger system preheats the air, syngas and water by utilizing the hot stream of gases produced by the burner. **Hotbox:** This is a big box where the fuel-cell stack, the heat exchanger system and the burner are placed. So, in addition to the heat exchanger system, the various components of the fuel-cell system are exchanging heat through radiative and convective transfer with the hotbox.

The overall system consists of 4 inputs that can be adjusted to achieve the desired performance of the system. There are several measurable outputs of the system and some unmeasurable outputs. The inputs and the outputs are described as following.

Inputs of the system

- q_{CH_4} the methane flowrate (L min^{-1})
- q_{air} the air flowrate (L min^{-1})
- I the current (A)
- q_{H_2O} the water flowrate (μg min^{-1})

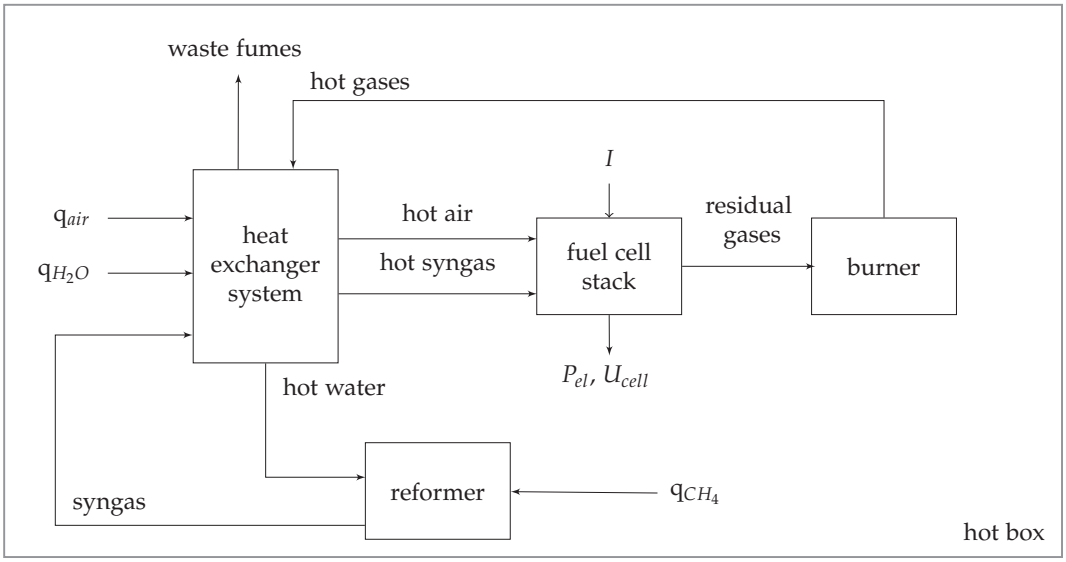


Figure 5.3 – Layout of the fuel-cell system.

Outputs of the system

- P_{el} the electrical power (W)
- U_{cell} the cell voltage (V)

Other measurable states of the system

- 15 temperatures of different components of the fuel-cell system.

Unmeasurable state of the system

- $\mathrm{q}_{fumes} = \sum_{i=1}^{3} \mathrm{q}_{fumes}^{HEX,i}$ the cold fumes. This is an output of the heat exchanger system that is unmeasurable and of no interest for the optimization.

Note that the power is a linear function of the current and the cell voltage

$$P_{el} = U_{cell} N_{cell} I \tag{5.18}$$

where N_{cell} is the number of cells in the stack and U_{cell} is a nonlinear function of the methane and the air flowrate.

Formulation of the Optimization Problem

The objective of the optimization is to maximize the steady-state efficiency of the system while providing the desired electrical power P_{el}^{ref}. The objective function reads

$$\eta := \frac{P_{el}}{LHV_{CH_4} \cdot \mathrm{q}_{CH_4}} \tag{5.19}$$

where LHV_{CH_4} is the lower heating value of the methane.

There are several constraints that are enforced while maximizing the efficiency. The equality constraint on the power demand P_{el}^{ref} is imposed as

$$P_{el} = P_{el}^{ref} \tag{5.20}$$

One of the major problem with the fuel cells is the ageing of the system. Indeed, it has a limited number of working cycles. In order to increase this number, the intensity use of the cell is decreased. This intensity can be quantified through fuel utilization FU, a dimensionless quantity, which is forced to stay below the value of 0.7:

$$FU := \frac{N_c}{8F}\frac{I}{n_{CH_4}} = \frac{N_c}{8F}\frac{I}{\tilde{v}_{CH_4}\mathrm{q}_{CH_4}} \leq 0.7 \tag{5.21}$$

where F is the Faraday constant, $n_{(.)}$ is a molar flow and $\tilde{v}_{(.)}$ is the conversion factor from mol to m^3. For the same reason, the cell voltage U_{cell} is constrained to be at least 0.7 (V).

To ensure complete reactions inside the reformer and the fuel-cell stack, excess ratios on the air and the water are set. These excess ratios are:

$$\lambda_w := \frac{1}{2}\frac{n_{H_2O}}{n_{CH_4}} = \frac{\tilde{v}_{H2O}}{\tilde{v}_{CH_4}} \cdot \frac{\mathrm{q}_{H_2O}}{\mathrm{q}_{CH_4}} \tag{5.22}$$

$$\lambda_{air} := \frac{1}{2}\frac{n_{air}}{n_{CH_4}} = \frac{\tilde{v}_{air}}{\tilde{v}_{CH_4}} \cdot \frac{\mathrm{q}_{air}}{\mathrm{q}_{CH_4}} \tag{5.23}$$

The constraint on the water excess ratio λ_w is set to be at least 1.5 and the air excess ratio λ_{air} is set to be at least 4.

In addition, the reformer temperature T_{ref} is constrained. It is set to lie between 663 (K) to 773 (K). As mentioned earlier, the input vector is $\mathbf{u} = [\mathrm{q}_{CH_4},\ \mathrm{q}_{air},\ I,\ \mathrm{q}_{H_2O},]^\mathsf{T}$.

The resulting optimization problem reads

$$\underset{\mathbf{u}}{\text{maximize}} \quad \eta = \frac{P_{el}}{LHV_{CH_4} \cdot \mathrm{q}_{CH_4}}, \tag{5.24a}$$

subject to

$$\begin{cases} P_{el} = P_{el}^{ref}, \quad P_{el}^{ref} \in \{50,\, 75\,100\}, \\ U_{cell} \geq 0.7\,(\mathrm{V}), \\ 390\,(^{\circ}\mathrm{C}) \leq T_{ref} \leq 500\,(^{\circ}\mathrm{C}), \end{cases} \quad \textit{constraints adapted via modifiers} \tag{5.24b}$$

$$\begin{cases} \boldsymbol{f}(\mathbf{u}) = \mathbf{0}, \\ FU \leq 0.7, \\ \lambda_{air} \geq 4, \\ \lambda_{w} \geq 1.5, \end{cases} \quad \textit{unadapted constraints} \tag{5.24c}$$

$$\begin{cases} 0.16\,(\mathrm{L\ min^{-1}}) \leq \mathrm{q}_{CH_4} \leq 0.6\,(\mathrm{L\ min^{-1}}), \\ 5\,(\mathrm{L\ min^{-1}}) \leq \mathrm{q}_{air} \leq 35\,(\mathrm{L\ min^{-1}}), \\ 0 \leq I \leq 50\,(\mathrm{A}), \\ 0\,(\mu\mathrm{g\ min^{-1}}) \leq \mathrm{q}_{H_2O} \leq 2000\,(\mu\mathrm{g\ min^{-1}}), \end{cases} \quad \textit{input bounds} \tag{5.24d}$$

where $\boldsymbol{f}(\mathbf{u}) = \mathbf{0}$ are the steady-state model equations. The system efficiency is maximized for 3 different power set points $P_{el}^{ref} = 50$ W, 75 W and 100 W. For the fuel-cell system, not all the constraints are adapted in the RTO schemes. The input bounds (5.24d) and the constraints (5.24c), except for the steady-state equality equations $\boldsymbol{f}(\mathbf{u})$, are the same for the plant and the model as their mappings to the inputs are perfectly known. Only the electrical power, the voltage inequality and the reformer temperature constraints are adapted via modifiers.

Structural Plant-Model Mismatch

In this simulation study, the *model* utilized for the application of different RTO methods consists of an ideal fuel-cell stack. That is, it is assumed that no potential losses occur inside the fuel cell. On the other hand, the simulated reality (plant) contains these potential losses. The plant-model mismatch reads

$$U_{cell} = U_N - \eta_{loss} \tag{5.25}$$

$$\eta_{loss} = \begin{cases} 0 & \text{Model} \\ \eta_{act,cath} + \eta_{diss} + \eta_{el} + \eta_{diff,an} + \eta_{diff,cath} + \eta_{MIC} & \text{Plant} \end{cases} \tag{5.26}$$

where U_N is the reversible cell voltage, η_{loss} is the sum of the overpotentials, $\eta_{act,cath}$ is the activation loss in the cathode, η_{diss} is the dissociation loss in the cathode, η_{el} is

number of inputs n_{u}	4
number of parameters n_θ	21
adapted inequality constraints	3
adapted equality constraint	1

Table 5.4 – Dimensionality of the fuel cell optimization problem.

the losses in the electrolyte, $\eta_{diff,an}$ is the diffusion loss in the anode, $\eta_{diff,cath}$ is the diffusion loss in the cathode, η_{MIC} is the ohmic loss in the metal interconnect. The modeling of these losses and the reversible cell voltage is given in [82].

The resulting model contains a total of 21 uncertain parameters. The uncertainty range is considered to be $\pm 15\,\%$ of the nominal value of the model parameters and the parametric uncertainty is assumed to be uniformly distributed in this range. Table 5.4 summarizes the dimensionality of the problem.

Different RTO schemes such as CA, MA, DMA and ADMA are applied to the simulated fuel-cell system. To this end, all the RTO schemes are started at the initial input point of $\mathbf{u}_0 = [0.3\,(\mathrm{L\ min})^{-1},\ 20\,(\mathrm{L\ min})^{-1},\ 25\,(\mathrm{A}),\ 600\,(\mu\mathrm{g\ min}^{-1})]^{\mathsf{T}}$. The initial value of P_{el}^{ref} is set at 50 W, then the value is sequentially set to 75 W and finally to 100 W. For all the RTO schemes, the input filter is applied and is set to $\boldsymbol{K} = diag(0.4\ 0.2\ 0.5\ 0.3)$. For evaluating the plant gradients/directional derivatives, a forward-finite-difference method is used with a small perturbation of 10^{-4} as only a noise-free case is considered. The noise in the measurements is treated in the later sections.

5.2.1 Application of CA

CA is applied to the fuel-cell system and the results are plotted in Figure 5.4. The y-axis of each of the plots in the figure represents either the objective function, the constraints, or the inputs. The x-axis of the plots is the total plant experiments performed, which in the case of CA is equal to the RTO iterations. The plant optimal values are plotted in red dashed-dotted lines.

The efficiency plot in Figure 5.4 shows that CA converges to a sub-optimal solution for each of the P_{el}^{ref} set points. CA correctly identifies the active constraints on the cell potential U_{cell} and the water excess ratio λ_w. It obviously meets the equality constraint on P_{el}^{ref}. However, it fails to identify the optimal values for the air flowrate q_{air} as it incorrectly goes to the upperbound on the air flowrate.

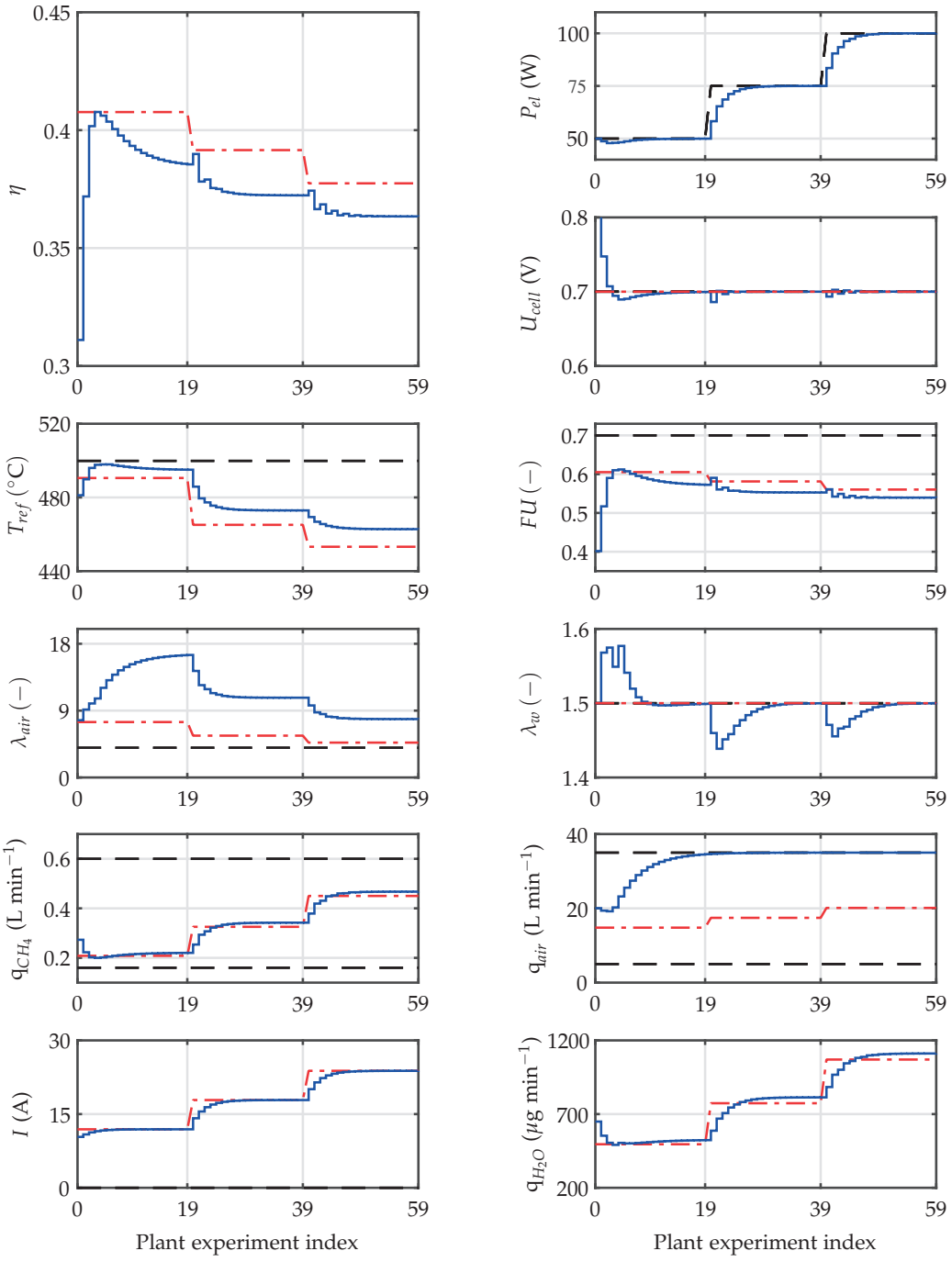


Figure 5.4 – Application of CA to the fuel-cell system for 3 different power set points. Red dashed-dotted line: plant optimum. Blue solid line: plant behavior. Black dashed line: constraint bounds.

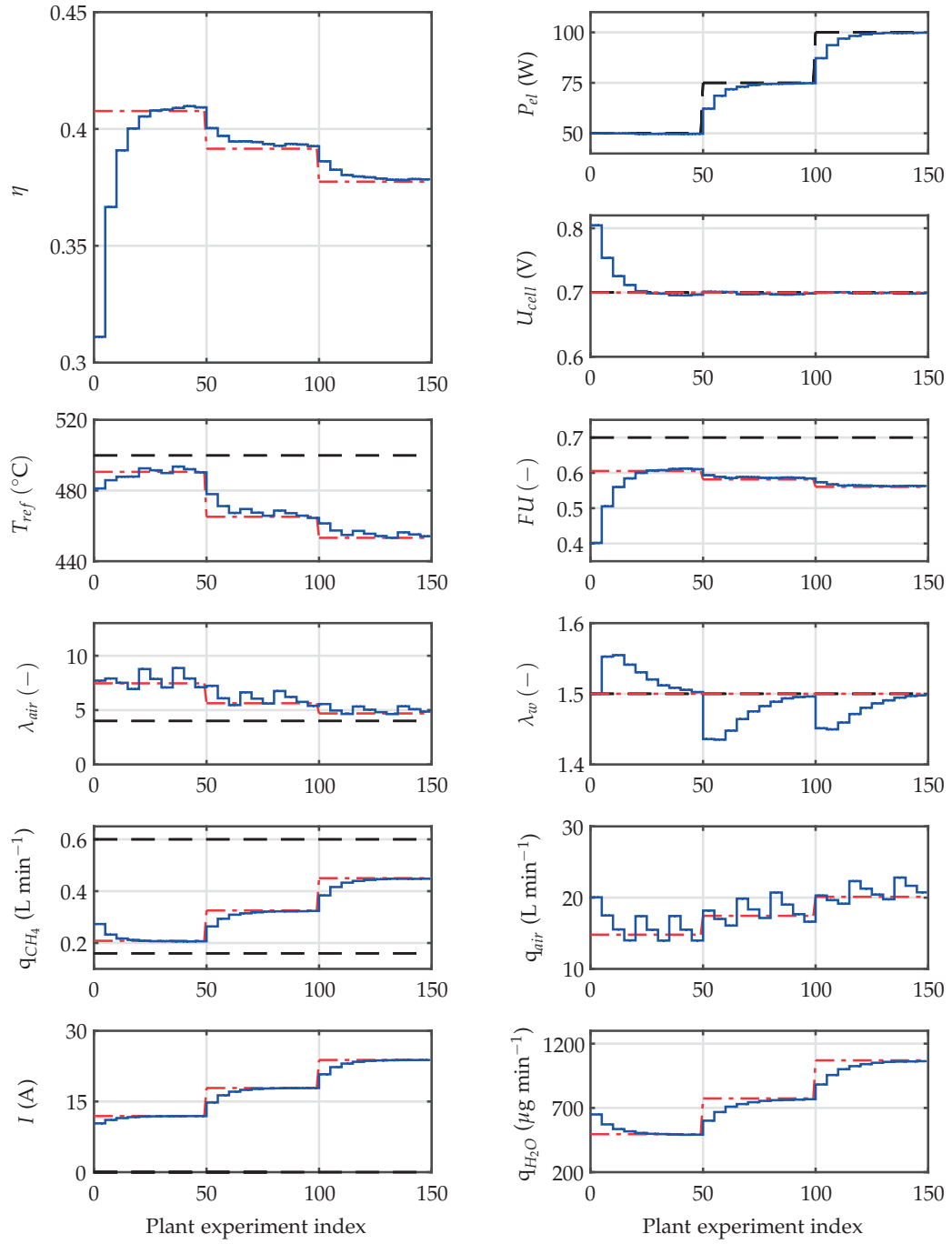


Figure 5.5 – Application of MA to the fuel-cell system for 3 different power set points. Red dashed-dotted line: plant optimum. Blue solid line: plant behavior. Black dashed line: constraint bounds.

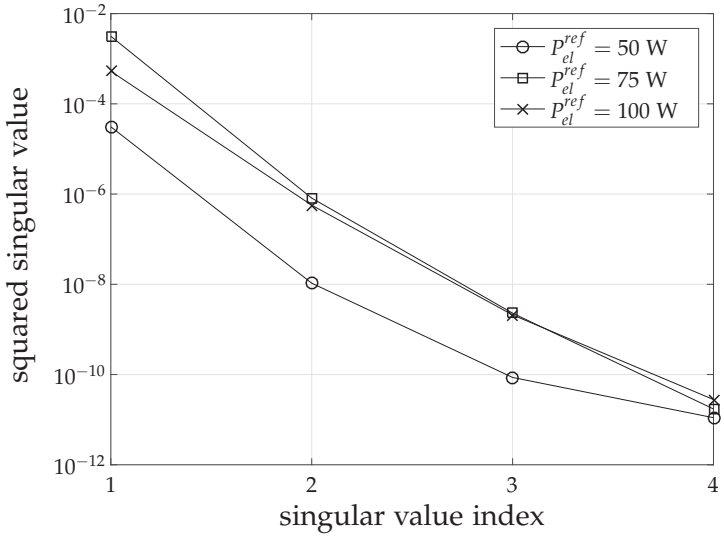


Figure 5.6 – Fuel-cell system: Singular values of the local sensitivity matrix in DMA.

5.2.2 Application of MA

CA indicates that, in order to reach plant optimality, gradient corrections are necessary. Thus, MA is applied to the fuel-cell system. The results are plotted in the Figure 5.5. The y-axis quantities in the plots remain the same. But the x-axis quantity - plant experiment index now also includes the gradient estimation experiments in addition to the RTO iterations.

The plots in Figure 5.5 show that the MA successfully finds the plant optimum. In addition to finding the correct set of active constraints, MA is able to find the optimal value for the air flowrate. However, this performance comes at the cost of excessive plant experiments. MA requires on average 30 plant experiments that includes experiments required for gradient estimation. Note that the air flowrate oscillates around the optimal value, although the efficiency remains more or less stable. This phenomenon occurs because the efficiency mapping to the air flowrate is relatively flat specially near the plant optimum. Therefore, the efficiency changes little upon large changes in the air flowrate.

Note that MA requires many more plant experiments to converge as compared to CA. This obviously happens due to additional plant experiments required for gradient estimation. The large number of plant experiments limits the use of MA for this application as the power setpoints may change at a faster rate then the time required for MA to converge. Therefore, to reduce the experimental burden of gradient estimation, DMA is tested next on the fuel-cell system.

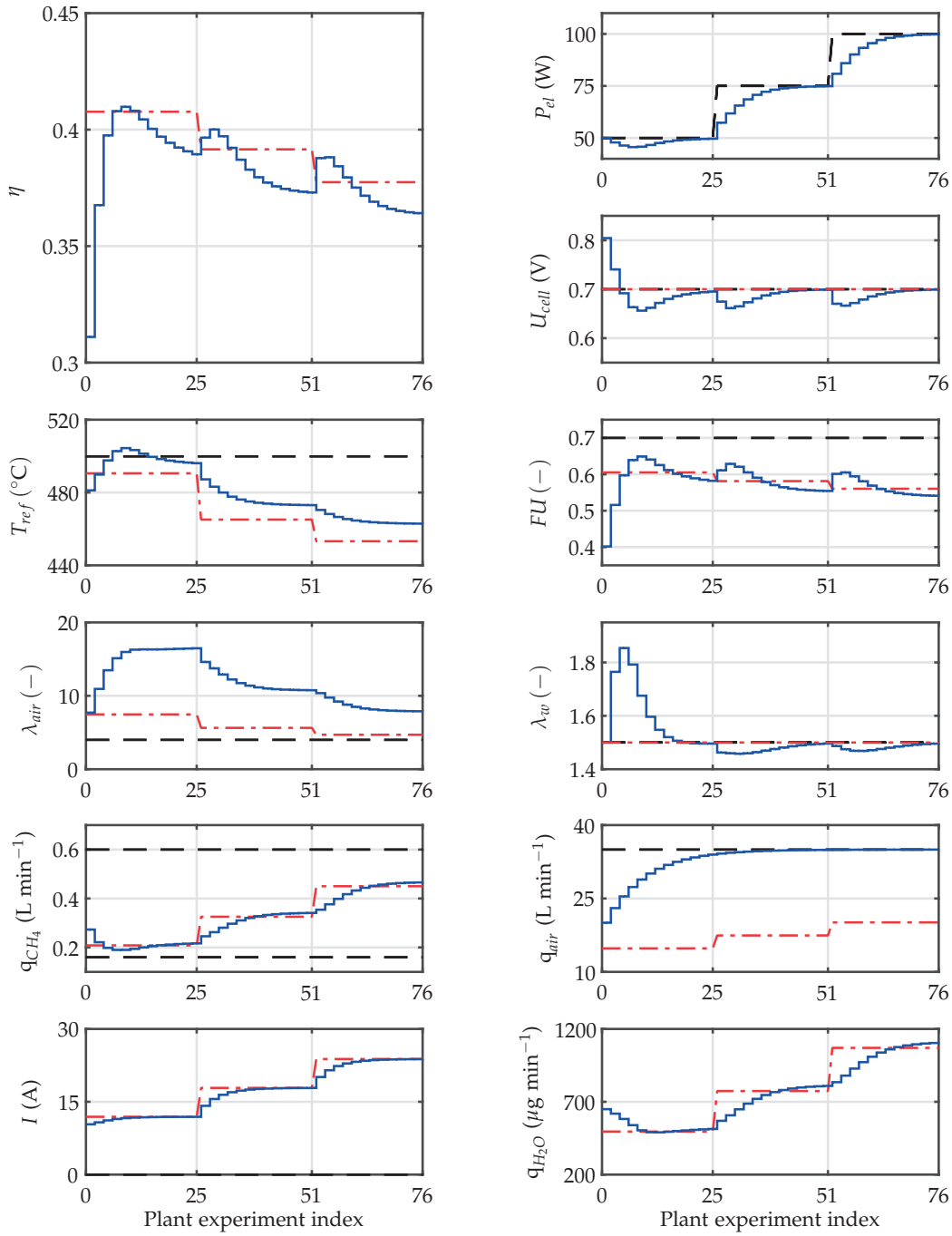


Figure 5.7 – Application of DMA to the fuel-cell system for 3 different power set points. Red dashed-dotted line: plant optimum. Blue solid line: plant behavior. Black dashed line: constraint bounds.

5.2.3 Application of DMA

In order to apply DMA, an offline local sensitivity analysis is carried out at the model optimums for three different P_{el}^{ref} values. The singular value decomposition of the local sensitivity matrix results in 4 singular directions in the input space. The corresponding squared singular values are plotted in the Figure 5.6. The large gap between the first and the second singular value reveals that there is only one privileged direction at each power setpoint P_{el}^{ref}.

In DMA, the privileged direction is fixed and, therefore, the plant directional derivative is estimated along the same privileged direction at each RTO iteration. The privileged direction found at $P_{el}^{ref} = 50$ W is $[0.4, 0.9, 0, 0.18]^{\mathsf{T}}$. Note that the constraint on the power P_{el} is tracked by the current I. Since, this is an equality constraint, the zeroth-order modifier is sufficient for its tracking and no gradient modifier is needed. Therefore, for the purpose of gradient estimation, the component corresponding to the current I is dropped from the privileged direction.

The results of DMA are plotted in the Figure 5.7. In terms of reaching maximum efficiency, DMA preforms similar to CA despite the directional-derivative corrections. Notice that 4 input components converge to the same value for CA and DMA at each power set-point P_{el}^{ref}. That is, DMA also fails to find the optimal value for the air flowrate q_{air} and the flowrate converges to its upperbound. This happens because the privileged direction found by the local sensitivity analysis is not the 'correct' direction. The local sensitivity analysis fails to capture the most sensitive input direction.

5.2.4 Application of ADMA

In ADMA, the privileged directions are computed via global sensitivity analysis. To this end, one can utilize Algorithm 3.2. In Algorithm 3.2, computing the sensitivity matrix $\hat{\boldsymbol{A}}_k$ via forward-finite-difference requires, at each random sample, $(n_{\mathrm{u}} + 1) \times (n_\theta + 1)$ model Lagrangian evaluations. This translates to 110 model Lagrangian evaluations. Each model Lagrangian evaluation takes about a fifth of a second. This implies that, if a total of 150 random samples are generated, then the computation of $\hat{\boldsymbol{A}}_k$ takes $(0.2) \cdot 150 \cdot 110 \approx 1$ h per RTO iteration. This computational time is prohibitive in this application as it is desired to apply RTO at a faster time scale, specially since the time constant of the fuel-cell-system dynamics is much faster than the computational time of one hour.

To reduce the computational effort, active subspaces are utilized. Active subspaces help in reducing the parameter space from 21 parameters to a lower number and, thus, in reducing the computation time of $\hat{\boldsymbol{A}}_k$. To discover active subspaces, a local-linear-model technique is used via Algorithm 3.5. The total number of samples used and other parameters used in Algorithm 3.5 are summarized in Table 5.5. The time for

Table 5.5 – Parameter values used in Algorithm 3.5 to discover active subspaces.

$\bar{N}$	150	M	60	n	75

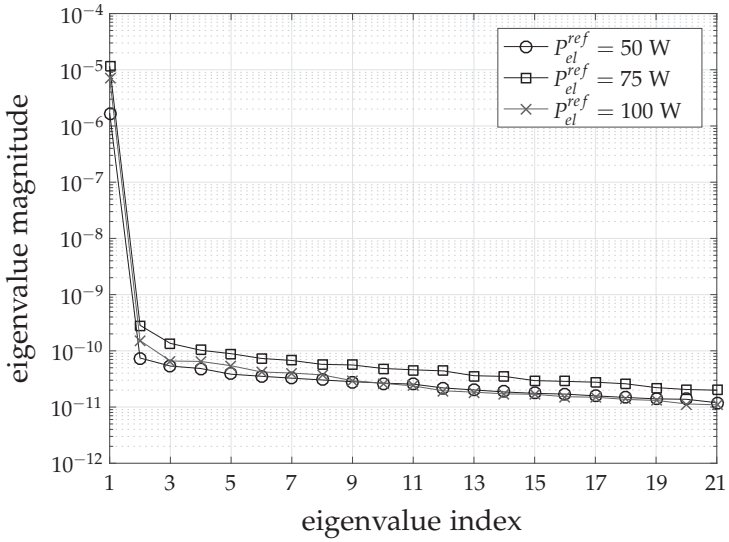


Figure 5.8 – Eigenvalues of $\hat{\boldsymbol{C}}_k$ for different power set points computed at the model optimum.

computing the active subspace is about 0.2(s) $\cdot$ 150 $\approx$ 30 s.

Active Subspaces for Dimension Reduction

For illustration purposes, the matrix $\hat{\boldsymbol{C}}_k$ is computed at the model optimum $(\mathbf{u}^\star, \boldsymbol{\mu}^\star)$ for each of the 3 power setpoints. The eigenvalues of $\hat{\boldsymbol{C}}_k$ are plotted in Figure (5.8). There is a large gap between the first and the second eigenvalues for each of the power setpoints. This indicates a presence of one dimensional active subspace. To confirm the presence of the active subspace, the sufficient summary plots (SSP)s introduced in Section 2.6.3 are constructed. The SSPs are plotted for the first 4 eigenvectors of $\hat{\boldsymbol{C}}_k$ corresponding to the 4 largest eigenvalues. The plots are shown in Figure 5.9. For each value of P_{el}^{ref}, a clear univariate trend appears for the SSP plot $(\mathcal{L}_k^{(j)}, (\hat{\mathbf{q}}_{1,k})^\mathsf{T}\boldsymbol{\theta}_j)$ corresponding to the first eigenvector, whereas for the rest of the eigenvectors the SSPs are scatter plots showing no particular trend. This confirms the presence of a single dimensional active subspace. Therefore, the parameter space is successfully reduced from $n_\theta = 21$ to $m = 1$.

An interesting trend to observe in the SSP plots is that the model Lagrangian is (approximately) linear in the active subspace. This implies that, with respect to the active subspace, the global sensitivity of the gradient of the model Lagrangian $(\nabla_\mathrm{u}\mathcal{L})$ can be approximated by the local sensitivity. Hence, only local sensitivity is computed

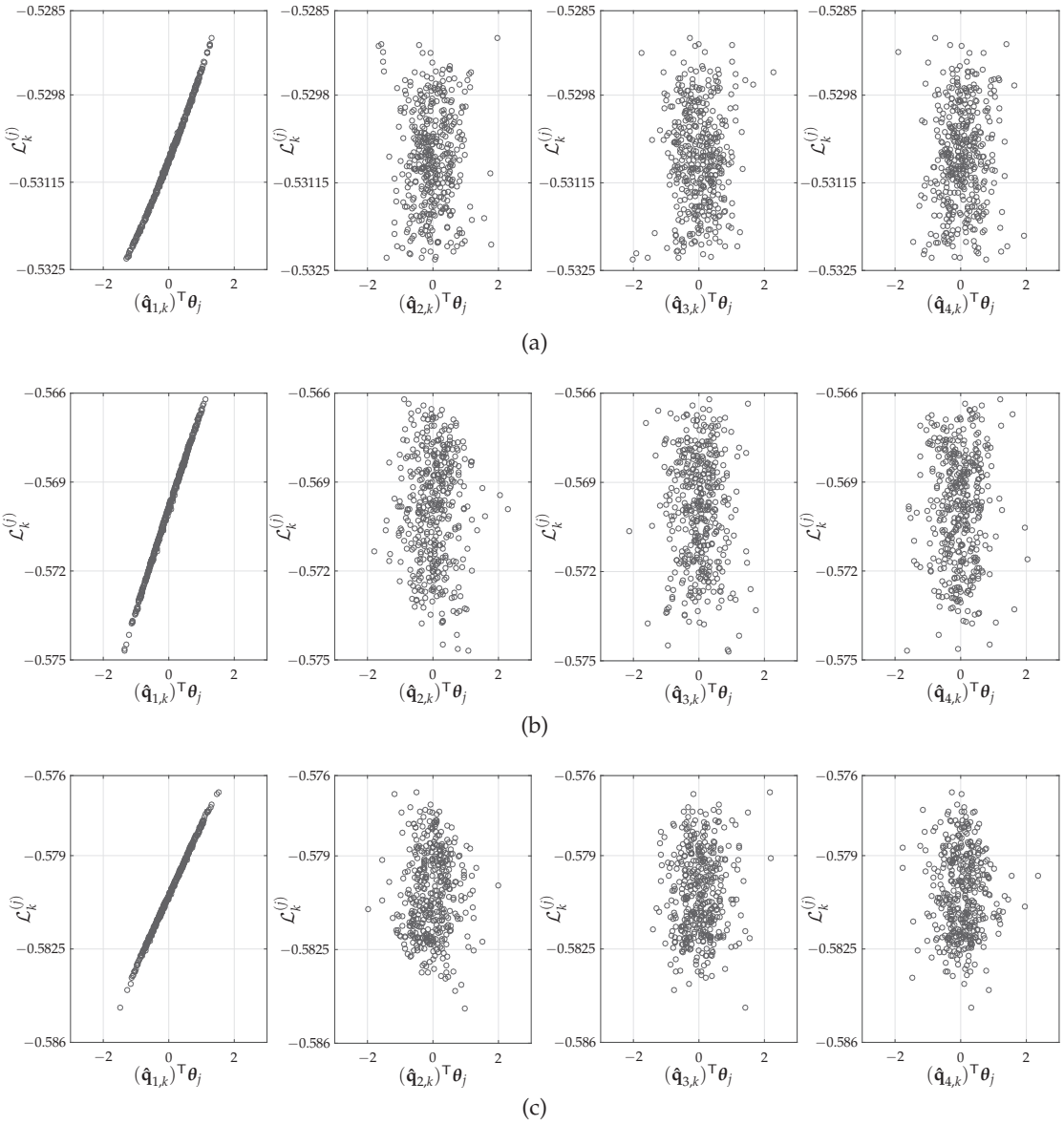


Figure 5.9 – Sufficient summary plots for active subspace discovery. (a) $P_{el}^{ref} = 50$ W. (b) $P_{el}^{ref} = 75$ W. (c) $P_{el}^{ref} = 100$ W.

to discover the privileged direction. It results in further reduction of the computational cost as the number of random samples N is replaced by $\bar{M} = 1$ in Algorithm 3.2. The overall reduction in the computational cost of the privileged direction is summarized in Table 5.6.

The results of active subspaces may seem to be surprising, as out of 21 parameters, only a single parameter direction is influential. This shows that models can be sloppy in the sense that there are many parametric redundancies. These redundancies are discovered through sensitivity analysis as done in active subspaces. Many para-

Table 5.6 – Summary of computational time taken by different algorithms for discovering privileged directions.

Algorithm	Evaluations of model Lagrangian	Computational time
Algorithm 3.2 with full parameter space	$(n_{\mathrm{u}}+1)\times(n_\theta+1)\times(N=150 \text{ samples})=16500$	1 hour
Algorithm 3.2 with active subspaces via Algorithm 3.5	$(n_{\mathrm{u}}+1)\times(m+1)\times(\bar{M}=1 \text{ sample})+\bar{N}=160$	32 seconds

Table 5.7 – Fuel-cell system: comparison of different RTO schemes.

RTO schemes	Optimality loss			Plant experiments required		
	$P_{el}^{ref}=50$ W	$P_{el}^{ref}=75$ W	$P_{el}^{ref}=100$ W	$P_{el}^{ref}=50$ W	$P_{el}^{ref}=75$ W	$P_{el}^{ref}=100$ W
CA	5.45 %	4.88 %	3.71 %	18	11	8
DMA	4.49 %	4.72 %	3.58 %	24	23	23
MA	0 %	0 %	0 %	24	30	36
ADMA	0.66 %	0.28 %	0.29 %	9	21	17

metric studies conducted using active subspaces have resulted in dramatic dimension reduction as documented in [28].

Recall that the observed linear trend in SSPs is still dependent on the value of power demand P_{el}^{ref}, the input vector $\mathbf{u}$ and the Lagrange multipliers vector $\boldsymbol{\mu}$. Indeed, even if one reduced parameter is sufficient, the linear combination to build it is not the same for different P_{el}^{ref} values. During the application of ADMA, similar linear trends with 1-dimensional active subspace are found at each RTO iteration as the values of P_{el}^{ref} and/or $(\mathbf{u}_k, \boldsymbol{\mu}_k)$ change from one iteration to the next. This confirms that, throughout the RTO iterations, the local sensitivity of $\nabla_{\mathrm{u}}\mathcal{L}$ with respect to the active subspace is capable of discovering the correct set of privileged directions as it adequately represents the global sensitivity.

Results and Discussion

Since a local sensitivity is computed with respect to a single parameter resulting from the active subspace, similar to condition (3.7) of DMA, the number of privileged directions is fixed at 1 (as $n_\theta \approx m = 1$). Note that, as detailed in Section 5.2.3 on DMA application to fuel-cell system, here too the component of the privileged direction corresponding to the current I is fixed at zero.

The results of ADMA application to the fuel-cell system are shown in Figure 5.10. The model gradient corrections in a single direction is sufficient in driving the plant to

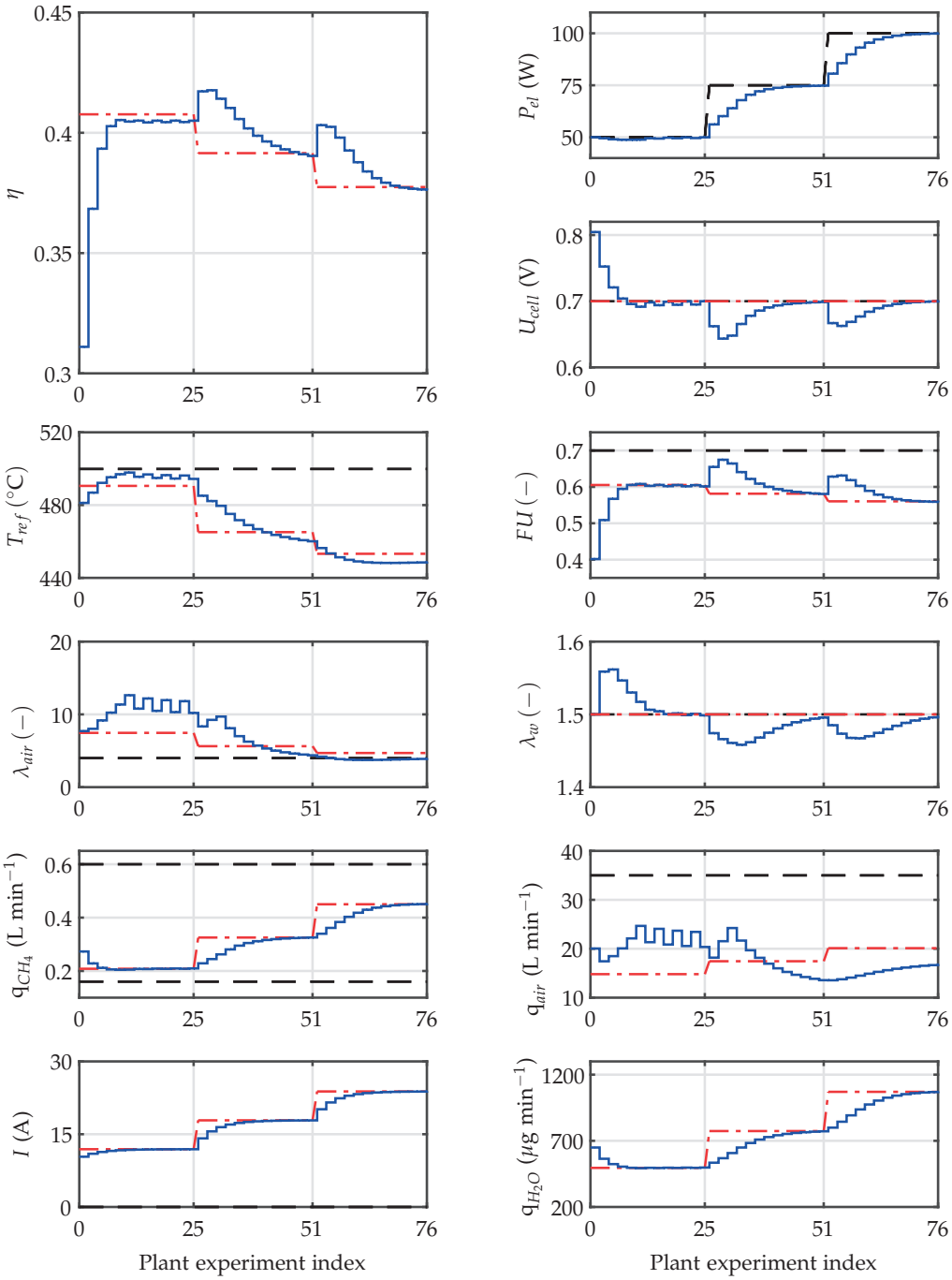


Figure 5.10 – Application of ADMA to the fuel-cell system for 3 different power set points. Red dashed-dotted line: plant optimum. Blue solid line: plant behavior. Black dashed line: constraint bounds.

(near) optimal efficiency. This confirms that the ADMA successful in finding the most sensitive input direction in which the model requires corrections.

The results obtained upon the application of different RTO schemes are summarized in Table 5.7. In terms of efficiency, MA and ADMA outperforms the other RTO schemes. ADMA results in negligible optimality losses whereas MA is truly optimal. However, when the total plant experiments required by the two RTO schemes are compared, ADMA clearly outperforms MA at all 3 power setpoints as it requires less than half the plant experiments required by MA. In fact, at power setpoint of 50 W, ADMA requires only a third of the plant experiments required by MA. Note that CA converges the fastest for all 3 power setpoints. However, CA results in considerable optimality loss as compared to ADMA. ADMA clearly outperforms the local-sensitivity-based DMA in both, optimality loss as well as in plant experiments required.

5.3 Summary

Optimal production of airborne wind energy via flying kites is heavily dependent on the path that the kite traverses. In the presence of uncertainties, such as plant-model mismatch and exogenous disturbances, it becomes essential to make measurement-based corrections to the underlying kite model. Thus, ADMA was tested in simulations on the kite system.

However, due to lack of high-order corrections in the model, ADMA fails to converge to the optimal solution. Instead, it results in an oscillatory behavior, where the plant first moves towards its optimum but then diverges due to disagreement in the model and the plant Hessian. Therefore, it is proposed to control the input step changes via a trust region. Trust region adjusts its size based on the agreement between the model prediction and the actual plant behavior. If needed, it shrinks itself to ensure that the model-plant agreement is maintained. Incorporation of the trust region in ADMA results in TR-ADMA algorithm that controls the input steps via trust region, which changes the algorithm behavior drastically, thus leading to excellent overall performance. The measurement noise is nicely handled by the combination of filtering the modifiers and trust-region adaptation.

This chapter further discusses the case study of a fuel-cell system. The aim of the study is efficiency maximization, while simultaneously ensuring the minimal system degradation by respecting critical, life-time enhancing process constraints. In addition, the constantly changing power demand pushes for a technological solution that can quickly adjust itself under varying conditions.

The MA-based RTO techniques are a natural fit to the problem as they guarantee post convergence constraint satisfaction via zeroth-order/bias correction terms and are capable of finding the varying optimal regimes caused by ever changing power demand conditions. Consequently, the simplest of all MA techniques, namely CA, is tested on the fuel-cell system. The simulation results show that CA converges to a fea-

sible yet sub-optimal point for every power setpoint. To overcome this sub-optimality, other variants of MA techniques have been tested. MA itself gives excellent performance in terms of feasibility and optimality, however, it takes too many plant experiments for (full) gradient estimation. In ever changing optimality conditions, too many plant experiments are prohibitive as MA may never converge under changing power demand.

To reduce the number of plant experiments, the local-sensitivity-based DMA is applied to the system. However, DMA performs similar to CA in terms of optimality which clearly indicates that offline-computed local sensitivities are not sufficient to point in right input (privileged) directions for model gradient corrections.

The above results clearly point towards the need for online-computed global sensitivity analysis that can reveal the correct set of privileged directions. However, computing global sensitivities via Algorithm 3.2 is very time consuming as the size of the parameter space is large. To reduce the parameter space, active subspaces are employed. The active subspaces are found by fitting computationally cheap locally-linear models that reveals a one-dimensional active subspace and, thereby, drastically reduces the parameter space from 21 parameters to a single parameter. Moreover, the plotted SSPs reveal an interesting linear trend between the model Lagrangian and the active subspace. This implies that local sensitivity with respect to the active subspace is enough to find the correct privileged direction. This results in a drastic reduction in computational time for finding the privileged directions. The time is reduced from one hour taken by Algorithm 3.2 when full parameter space is employed to 32 s when Algorithm 3.2 exploits the active subspace and, thus, finds sensitivities only locally in only a single parameter direction.

The privileged directions in ADMA are recomputed at each RTO iteration and, when applied to the fuel-cell system, they give excellent results. Not only near-optimal performance is achieved as opposed to CA and DMA, but also the total number of plant experiments required is significantly reduced when compared to MA.

6 Conclusion

6.1 Summary

This thesis shows that a tandem between the optimality conditions and the model parameters naturally occurs, which can be unearthed when the right tools are put to work. As any naturally occurring phenomenon, this tandem has its limitations that deprives parameter adaptation from independent KKT matching. Such a model deficit can be well compensated by the tailor-made modifiers leading to perfect KKT matching. On the other hand, when this limitation is rather seen as a strength and more confidence is shown in the models, it leads to methodologically chosen experiments that pushes towards the plant NCOs.

The first RTO scheme that is proposed here addresses the MA's requirement of excessive plant experiments for gradient estimation that may be prohibitive in practice. The experimental cost can be reduced by estimating the gradient only along a few privileged input directions found via offline local parametric sensitivity analysis. Unfortunately, the sensitivity analysis preformed at the model optimum is often no more valid as the input iterates move away from the model optimum and the active set of constraints change. Moreover, local sensitivity is weak in approximating the corresponding global phenomenon when the models are highly nonlinear in parameters.

The proposed ADMA methodology tackles these issues by conducting a *global* sensitivity analysis performed *online*. The active subspace theory is tailored to reveal the privileged directions by computing the sensitivity of the model Lagrangian gradient at random parameter samples. Each sensitivity sample is augmented to a big matrix that represents global sensitivities. The eigenvalue decomposition of the sensitivity matrix uncovers the input directions where the gradients are highly sensitive to parametric perturbations. This procedure is repeated at each RTO iteration as the inputs and the active set of constraints change. The modifiers are adapted to match the model and the plant gradients along the privileged directions.

The simulation studies of run-to-run optimization of semi-batch processes revealed that only a few privileged directions found via global sensitivity analysis are sufficient to reach optimal plant performance even in the presence of structural plant-model mismatch. Hence, fewer plant experiments are performed to estimate plant gradients along the handful of privileged input directions.

Certain applications may have faster time scales and, therefore, the computational time for global sensitivity analysis poses a problem. This challenge is handled by again altering the active subspace tools so that parametric sensitivity of the model Lagrangian gradient is approximated by only computing the Lagrangian gradients at random parameter samples. The subtraction of mean from the samples approximates the Lagrangian gradient sensitivity quite well and results in large reduction in computational time. Alternatively, when the parametric dimension is large, active subspace theory is directly employed to find a low-dimensional influential parameter direction also referred to as an active subspace. The sensitivities are then computed only with respect to the low-dimensional active subspace of parameters that drastically reduces the computational cost.

This thesis further discusses the role of model parameters and modifiers in influencing the KKT conditions. It is found that the structural independence is an inherent property of modifier adaptation that enables independent KKT matching. The same structural independence is searched in a given model that leads to the concept of mirror parameters. Mirror parameters are the model parameters that behave similar to modifiers in terms of their ability to independently influence the KKT conditions. The presence of mirror parameters is discovered by finding the influential and non-influential parameter spaces of each of the KKT element. This reveals whether there exists a one-to-one correspondence between parameters and KKT elements. If such a correspondence exists, then each KKT element of the model can be matched with that of the plant by adjusting the corresponding mirror parameter. If some KKT elements lack mirror parameters, then the modifiers are added to compensate. Moreover, if a gradient component and a constraint compete for the same influential parameter space, since the gradient estimation is expensive, the gradient component is given priority and it is matched by adapting the influential space. The constraint is then paired with a zeroth-order modifier.

The above discussed framework of generalized model adaptation is further developed to find the privileged direction to influential parameter correspondence. This translates to reducing the plant suboptimality at a reasonable experimental cost as only partial KKT matching is performed. The simulation study of steady-state optimization of the Williams-Otto process shows that the tandem between the privileged direction and the corresponding influential parameter is highly successful in reaching near-optimal plant performance despite structural plant-model mismatch and the presence of noise.

Finally, the developed ADMA algorithm is tested on advanced simulation studies of airborne wind energy system and solid-oxide fuel-cell system. For the case study of a simulated flying kite, ADMA without a trust region is not able to converge as the affine local corrections are insufficient to capture the plant behavior at far away input points. Such a problem is overcome by controlling the input steps via trust-region, which changes the algorithm behaviour drastically,thereby leading to an excellent overall performance.

The simulation study of the fuel-cell system is used to test different MA-based methods, where the goal is to maximize the efficiency of the system while satisfying lifetime critical process constraints. When only constraints are adapted using CA, the system performance improves from its nominal values but converges to a suboptimal but feasible input value. DMA performs more or less the same as CA and converges also to a suboptimal input value, which shows that the offline-computed local sensitivity analysis is insufficient in discovering the correct privileged input directions. MA obviously converges to the optimal value, however, if the process conditions change at a higher frequency than the one at which MA converges, then MA may never be able to reach the plant optimality. The convergence speed can be improved by reducing the number of plant experiments required to reach plant optimality. To this end, ADMA is applied that finds privileged directions via global sensitivity analysis that are computed online. Thereby, ADMA converges to near-optimal performance, while requiring considerably less experiments than MA.

6.2 Perspectives

This thesis leaves many open research problems that are essential for the techniques developed here to become industrial practice. In addition, the developed methodologies can be applied to other domains that deal with maximizing the information in minimum effort.

Alternate Methods for Global Sensitivity Analysis

This thesis shows that global sensitivity methods can outperform local sensitivities in computing the most relevant set of privileged directions. To this end, the ideas of active subspaces are tailored to RTO needs. However, the literature is full of alternative methodologies for conducting global sensitivity analysis [25, 63, 112]. For instance, it would be interesting if Sobol sensitivity indices [124] are utilized for finding influential parameters and compared with the results obtained via active subspaces.

Moreover, further research is needed to develop more efficient methods to compute influential parameters and privileged input directions. For instance, Morris screening

methods [77, 91] can be utilized to compute the sensitivities instead of performing finite-differences on random parameter samples. Another potential solution is to construct surrogate models for faster computations by using response surface models, gaussian processes [13, 99, 128] or polynomial chaos expansions [36] or radial basis functions [98].

Privileged Surfaces

Privileged input directions constitute linear combinations of inputs that best describe the gradient variability upon parametric perturbations. It should be investigated if, instead of a linear combination, a response surface [13] could be found such that the Lagrangian gradient varies significantly along the surface when the parameters are varied. The goal is then to use measurements to learn the plant mapping on this surface. The learned response surface can be utilized to compensate for plant-model mismatch. Ideally, response surface should be designed such that minimal experiments are required to learn the plant behavior along the surface. Privileged surfaces would be really helpful in the cases when the historical plant data is available that is sufficiently rich in information. Such data can be utilized in the construction of privileged surfaces.

Closed-Loop Implementation

The methodologies developed here are designed by considering the manipulated variables as the decision variables (referred to as input $\mathbf{u}$). However, the process systems in industry run in closed-loop, where the decision variables are the set-points passed to a controller that manipulates the input $\mathbf{u}$.

A straightforward solution is to directly use set-points as decision variables in numerical optimization. However, this requires that the controller is robustly designed to work at each set-point passed by the optimization layer. The controllers are designed to work mostly around the local operating conditions and may not be appropriate when there are large set-point changes. Hence, there is a need for integrating the RTO and control design techniques for practical implementation of RTO methods, such as ADMA. To this end, robust control design techniques [71, 107, 122] would be very useful that can handle changing operating conditions.

An alternative is to combine implicit and explicit RTO schemes. An implicit RTO scheme such as NCO tracking [46, 125, 127] constructs a solution model via offline numerical optimization. The solution model is used to design the control structure. The performance of the scheme is dependent on the solution model.

To update the solution model, global-sensitivity-based privileged directions can

be utilized for system excitation and process model can be updated by the resulting plant measurements. This could be done by either adapting modifiers or influential parameters. The process model can then be utilized to update the solution model via numerical optimization.

Adaptive Control

In adaptive control, model parameters are adapted and the updated model is used in the feedback control design. The goal is to find and adapt the model parameters that influences the controller performance the most and thereby, adjust the control law to achieve the desired behavior [75, 129]. Clearly, the framework developed here fits the adaptive control well. The main difference is that the optimization objectives are replaced by the control objectives.

Optimal Experimental Design in Systems Biology

The models constructed for predicting cellular behavior can be calibrated by optimal experimental design via dynamic experimental techniques [14, 106]. Carefully designed dynamic perturbations excites the real process to extract information that reduces the variance of parameter estimates by manyfold when compared to intuition-driven non-optimized experiments [7]. To this end, optimal control problems are formulated that yield most promising temporal input profiles for system excitation [9]. A tandem between the excitation input signal and parameter estimation is established that is analogous to the synergy found between privileged input directions and mirror parameters. Hence, the discovered knowledge in this thesis can be applied to better predict the behavior of cellular systems via iterative experiments that utilize optimized inputs.

Optimal Control of Drug Therapy in Cancer Treatment

Cancer remains one of the most dangerous illnesses that causes many deaths every year. The traditional treatment regimes include administering chemotherapy. The dosage of the therapy must be carefully adjusted in order to cause the minimum damage to healthy tissue, whilst killing a maximum number of tumour cells [92]. Recently, model-based optimal treatment regimes are proposed that can minimize the cancer cells in addition to the total amount of chemotherapy [40, 64]. However, the model calibration is patient dependent [40]. Therefore, only minimal experiments can be performed, if any. Hence, it becomes important to understand well the relation between drug regimes and the parameter identifiability, which directly links to the developments in this thesis. The mirror parameters can be found that are adapted for each patient while ensuring minimal experimentation via system excitation along

privileged input directions.

A Models Studied in Chapter 3

A.1 Williams-Otto Reactor Model

The model equations are as follows:

$$\frac{\mathrm{d}V}{\mathrm{d}t} = \frac{F_B}{q} \tag{A.1a}$$

$$\frac{\mathrm{d}m_A}{\mathrm{d}t} = -r_1 V \tag{A.1b}$$

$$\frac{\mathrm{d}m_B}{\mathrm{d}t} = F_B - \frac{M_{w_A}}{M_{w_B}} r_1 V + r_2 V \tag{A.1c}$$

$$\frac{\mathrm{d}m_C}{\mathrm{d}t} = \frac{M_{w_C}}{M_{w_A}} r_1 V - \frac{M_{w_C}}{M_{w_B}} r_2 V - r_3 V \tag{A.1d}$$

$$\frac{\mathrm{d}m_E}{\mathrm{d}t} = \frac{M_{w_E}}{M_{w_B}} r_2 V \tag{A.1e}$$

$$\frac{\mathrm{d}m_G}{\mathrm{d}t} = \frac{M_{w_G}}{M_{w_C}} r_3 V \tag{A.1f}$$

$$\frac{\mathrm{d}m_P}{\mathrm{d}t} = \frac{M_{w_P}}{M_{w_B}} r_2 V - \frac{M_{w_P}}{M_{w_C}} r_3 V \tag{A.1g}$$

$$\frac{\mathrm{d}T_r}{\mathrm{d}t} = \frac{H}{V C_p} \tag{A.1h}$$

$$c_i = m_i / V; \quad i = A, B, C, P, E, G$$

$$k_i = a_i e^{-b_i (T_r + T_{ref})}; \quad i = 1, 2, 3$$

$$r_1 = k_1 c_A c_B; \quad r_2 = k_2 c_B c_C; \quad r_3 = k_3 c_C c_P$$

$$H = F_B C_p T_{in} - \Delta H_1 r_1 V - \Delta H_2 r_2 V - \Delta H_3 r_3 V - V \frac{A_0}{V_0} U (T_r - T_w)$$

Table A.1 – Reaction parameters and operating conditions for the Williams-Otto semi-batch reactor

Variable	Definition	Value
M_A, M_B, M_P	Molar mass - components A, B, P	100 kg kmol^{-1}
M_C, M_E	Molar mass - component C, G	200 kg kmol^{-1}
M_G	Molar mass - component G	300 kg kmol^{-1}
a_3	Pre-exponential fraction - reaction 3	$2.6745 \cdot 10^{12}$ s^{-1}
b_3	Activation energy - reaction 3	11111 K
T_{ref}	Reference temperature	273.15 K
T_{in}	Inlet temperature (B)	308.15 K
ΔH_1	Enthalpy - reaction 1	236.8 kJ kg^{-1}
ΔH_2	Enthalpy - reaction 2	158.3 kJ kg^{-1}
ΔH_3	Enthalpy - reaction 3	226.3 kJ kg^{-1}
A_0	Heat-transfer area	9.2903 m^2
V_0	Cooling jacket volume	2.1052 m^3
U	Heat-transfer coefficient	0.23082 kJ (m^2 K s)$^{-1}$
$V(0)$	Initial reactor volume	2 m^3
$T_r(0)$	Initial reactor temperature	338.15 K
$m_A(0)$	Initial mass - component A	2000 kg
$m_B(0), m_C(0), m_P(0),$ $m_E(0), m_G(0)$	Initial mass - components B, C, P, E, G	0 kg
C_p	Specific heat capacity	4.184 kJ kg^{-1} C^{-1}
q	Fluid density	1000 kg m^3
P_P	Price of P	555.4 \$ kg^{-1}
P_E	Price of E	125.91 \$ kg^{-1}

A.2 Diketene-Pyrrole Reactor Model

The first-principles model for the semi-batch reactor reads:

$$\frac{\mathrm{d}c_A}{\mathrm{d}t} = -\mathrm{k}_1 c_A c_B - \frac{F_B}{V} c_A \tag{A.2a}$$

$$\frac{\mathrm{d}c_B}{\mathrm{d}t} = -\mathrm{k}_1 c_A c_B - 2\mathrm{k}_2 c_B^2 - \mathrm{k}_3 c_B - \mathrm{k}_4 c_B c_C + \frac{F_B}{V}(c_B^{in} - c_B) \tag{A.2b}$$

$$\frac{\mathrm{d}c_C}{\mathrm{d}t} = \mathrm{k}_1 c_A c_B - \mathrm{k}_4 c_B c_C - \frac{F_B}{V}(c_C) \tag{A.2c}$$

$$\frac{\mathrm{d}c_D}{\mathrm{d}t} = \mathrm{k}_2 c_B^2 - \frac{F_B}{V}(c_D) \tag{A.2d}$$

$$\frac{\mathrm{d}V}{\mathrm{d}t} = F_B, \tag{A.2e}$$

where c_A, c_B, c_C and c_D represent the concentrations of the species A, B, C and D, respectively. V is the reactor volume, F_B is the inlet flowrate of species B, and c_B^{in} is the concentration of B in the feed.

Table A.2 – Reaction parameters and operating conditions for the diketene-pyrrole semi-batch reactor

Variable	Definition	Value
c_B^{in}	Concentration of B in the feed	5 mol L^{-1}
$V(0)$	Initial reactor volume 0	1 L
$c_A(0)$	Initial concentration of A	0.72 mol L^{-1}
$c_B(0)$	Initial concentration of B	0.05 mol L^{-1}
$c_C(0)$	Initial concentration of C	0.08 mol L^{-1}
$c_D(0)$	Initial concentration of D	0.01 mol L^{-1}
t_f	Final time	250 min
F_B^{max}	Maximal inlet flowrate	$2 \cdot 10^{-3}$ L min^{-1}
c_B^{max}	Maximal concentration of B at final time	0.025 mol L^{-1}
c_D^{max}	Maximal concentration of D at final time	0.15 mol L^{-1}
ω	Weight	10 mol min L^{-2}

B Model Studied in Chapter 4

B.1 Williams-Otto Process

B.1.1 Plant Equations

The different units of the plant are described. The steady-state component material balance and the reaction equations around the reactor are given as:

$$F_A + F_{T,A} - F_{R,A} - V\, r_1 = 0, \tag{B.1a}$$

$$F_B + F_{T,B} - F_{R,B} - V\, r_1 - V\, r_2 = 0, \tag{B.1b}$$

$$F_{T,C} - F_{R,C} + \frac{M_C}{M_A}\, V\, r_1 - \frac{M_C}{M_B}\, V\, r_2 - V\, r_3 = 0, \tag{B.1c}$$

$$F_{T,E} - F_{R,E} + \frac{M_C}{M_B}\, V\, r_2 = 0, \tag{B.1d}$$

$$F_{T,G} - F_{R,G} + \frac{M_G}{M_C}\, V\, r_3 = 0, \tag{B.1e}$$

$$F_{T,P} - F_{R,P} + \frac{M_P}{M_B}\, V\, r_2 - \frac{M_P}{M_C}\, V\, r_3 = 0, \tag{B.1f}$$

$$r_1 = \mathsf{k}_1\, \frac{F_{R,A}\, F_{R,B}}{(F_R)^2}; \quad r_2 = \mathsf{k}_2\, \frac{F_{R,B}\, F_{R,C}}{(F_R)^2}; \quad r_3 = \mathsf{k}_3\, \frac{F_{R,C}\, F_{R,P}}{(F_R)^2}, \tag{B.1g}$$

$$F_R - (F_{R,A} + F_{R,B} + F_{R,C} + F_{R,E} + F_{R,G} + F_{R,P}) = 0, \tag{B.1h}$$

$$\mathsf{k}_i = \mathsf{A}_i\, \exp\left(\frac{-\mathsf{E}_i}{T_r + 273.15}\right); \quad i = 1, 2, 3. \tag{B.1i}$$

Table B.1 – Williams-Otto reactor: Plant parameters and parameters that are common to both the plant and the model

Plant parameter	Value	Common parameter	Value	Common parameter	Value
A_1 (s^{-1})	$1.6599 \cdot 10^6$	V (kg)	2105	P_A	0.0441
A_2 (s^{-1})	$7.2117 \cdot 10^8$	M_A	1	P_B	0.0661
A_3 (s^{-1})	$2.6745 \cdot 10^{12}$	M_B	1	P_D	0.015
E_1 (K)	6666.7	M_C	2	P_G	0.022
E_2 (K)	8333.3	M_P	1	P_P	0.6614
E_3 (K)	11111	M_E	2	P_R	4.8943
β	0.1	M_G	3	-	-

The decanter unit assumes perfect separation between the product G and the rest of the components. The material balance for the decanter reads:

$$F_{S,i} - F_{R,i} = 0, \quad i = A, B, C, E, P, \tag{B.2a}$$
$$F_{S,G} = 0, \tag{B.2b}$$
$$F_{G,i} = 0, \quad i = A, B, C, E, P, \tag{B.2c}$$
$$F_{G,G} - F_{R,G} = 0. \tag{B.2d}$$

The distillation unit assumes a pure separation of product P overhead but also assumes that some of the product is retained in the bottoms due to formation of an azeotrope (the fraction β of the mass flowrate of component E in the column feed is taken as the amount of P retained in the bottoms). The material balance reads:

$$F_{Y,i} - F_{S,i} = 0, \quad i = A, B, C, E, \tag{B.3a}$$
$$F_{P,i} = 0, \quad i = A, B, C, E, G, \tag{B.3b}$$
$$F_{P,P} - F_{S,P} + \beta F_{S,E} = 0, \tag{B.3c}$$
$$F_{Y,P} - \beta F_{S,E} = 0, \tag{B.3d}$$
$$F_{Y,G} = 0, \tag{B.3e}$$

The splitter unit equations can be written as:

$$F_{T,i} - \alpha F_{Y,i} = 0, \quad i = A, B, C, E, P, \tag{B.4a}$$
$$F_{D,i} - (1 - \alpha) F_{Y,i} = 0, \quad i = A, B, C, E, P. \tag{B.4b}$$

The parameter values for the plant are give in Table B.1.

Table B.2 – Williams-Otto reactor: Model parameters

Model parameter	Nominal parameter value $\boldsymbol{\theta}_0$	Uncertainty range	Probability distribution
$\bar{A}_1$ (s^{-1})	$1.0728 \cdot 10^8$	$[8.5824 \cdot 10^7, 1.2874 \cdot 10^8]$	uniform
$\bar{A}_2$ (s^{-1})	$1.4155 \cdot 10^{13}$	$[1.1324 \cdot 10^{13}, 1.6986 \cdot 10^{13}]$	uniform
$\bar{E}_1$ (K)	7458.6	[5966.88, 8950.32]	uniform
$\bar{E}_2$ (K)	13789.5	[11031.6, 16547.4]	uniform
β	0.075	[0.06, 0.09]	uniform

B.1.2 Model Equations

The material balance and reaction equations around the reactor for the two-reaction model read:

$$F_A + F_{T,A} - F_{R,A} - V\,\bar{r}_1 - V\,\bar{r}_2 = 0, \tag{B.5a}$$

$$F_B + F_{T,B} - F_{R,B} - 2\,V\,\bar{r}_1 - V\,\bar{r}_2 = 0, \tag{B.5b}$$

$$F_{T,E} - F_{R,E} + \frac{M_E}{M_A}\,V\,\bar{r}_1 = 0, \tag{B.5c}$$

$$F_{T,G} - F_{R,G} + \frac{M_G}{M_A}\,V\,\bar{r}_2 = 0, \tag{B.5d}$$

$$F_{T,P} - F_{R,P} + \frac{M_P}{M_A}\,V\,\bar{r}_1 - \frac{M_P}{M_A}\,V\,\bar{r}_2 = 0, \tag{B.5e}$$

$$\bar{r}_1 = \bar{k}_1\,\frac{F_{R,A}\,(F_{R,B})^2}{(F_R)^3}; \quad \bar{r}_2 = \bar{k}_2\,\frac{F_{R,A}\,F_{R,B}\,F_{R,P}}{(F_R)^3}, \tag{B.5f}$$

$$F_R - (F_{R,A} + F_{R,B} + F_{R,E} + F_{R,G} + F_{R,P}) = 0, \tag{B.5g}$$

$$\bar{k}_i = \bar{A}_i\,\exp\left(\frac{-\bar{E}_i}{T_r + 273.15}\right); \quad i = 1, 2. \tag{B.5h}$$

For the other units, the model has the same material balance equations as those for the plant. Of course, the intermediate species C is dropped from the equations. The adjustable parameter values are given in Table B.2.

Bibliography

[1] H. S. Abdel-Khalik, Y. Bang, and C. Wang, *Overview of hybrid subspace methods for uncertainty quantification, sensitivity analysis*, Annals of Nuclear Energy, 52 (2013), pp. 28 – 46.

[2] A. Ahmad, W. Gao, and S. Engell, *Effective model adaptation in iterative rto*, in 27th European Symposium on Computer Aided Process Engineering, vol. 40 of Computer Aided Chemical Engineering, Elsevier, 2017, pp. 1717 – 1722.

[3] V. Alstad and S. Skogestad, *Null space method for selecting optimal measurement combinations as controlled variables*, Ind. Eng. Chem. Res., 46 (2007), pp. 846–853.

[4] J. Andersson, *A General-Purpose Software Framework for Dynamic Optimization*, PhD thesis, Arenberg Doctoral School, KU Leuven, Department of Electrical Engineering and Optimization in Engineering Center, Kasteelpark Arenberg 10, 3001-Heverlee, Belgium, October 2013.

[5] C. L. Archer, *An Introduction to Meteorology for Airborne Wind Energy*, Springer Berlin Heidelberg, 2013, pp. 81–94.

[6] K. B. Ariyur and M. Krstic, *Real-Time Optimization by Extremum-Seeking Control*, New York: John Wiley & Sons, 2003.

[7] S. Bandara, J. P. Schlöder, R. Eils, H. G. Bock, and T. Meyer, *Optimal experimental design for parameter estimation of a cell signaling model*, PLOS Computational Biology, 5 (2009), pp. 1–12.

[8] Y. Bang, H. S. Abdel-Khalik, and J. M. Hite, *Hybrid reduced order modeling applied to nonlinear models*, Int. J. Numer. Meth. Engng., 91 (2012), pp. 929–949.

[9] I. Bauer, H. G. Bock, S. Körkel, and J. P. Schlöder, *Numerical methods for optimum experimental design in dae systems*, J. Computational and Applied Mathematics, 120 (2000), pp. 1 – 25.

[10] M. S. Bazaraa, H. D. Sherali, and C. M. Shetty, *Nonlinear Programming: Theory and Algorithms*, New York: John Wiley & Sons, 3rd ed., 2006.

[11] M. O. Besenhard, A. Chaudhury, T. Vetter, R. Ramachandran, and J. G. Khinast, *Evaluation of parameter estimation methods for crystallization processes modeled via population balance equations*, Chemical Engineering Research and Design, 94 (2015), pp. 275 – 289.

[12] L. Biegler, Y. Lang, and W. Lin, *Multi-scale optimization for process systems engineering*, Comp. Chem. Engng., 60 (2014), pp. 17–30.

[13] G. E. Box and N. R. Draper, *Response surfaces, mixtures, and ridge analyses*, vol. 649, New York: John Wiley & Sons, 2007.

[14] N. Braniff and B. Ingalls, *New opportunities for optimal design of dynamic experiments in systems and synthetic biology*, Current Opinion in Systems Biology, 9 (2018), pp. 42 – 48. Mathematic modelling.

[15] M. Brdys and P. Tatjewski, *An algorithm for steady-state optimising dual control of uncertain plants*, in Proc IFAC Workshop on New Trends in Design of Control Systems, 1994, pp. 249–254.

[16] M. Brdyś and P. Tatjewski, *Iterative Algorithms for Multilayer Optimizing Control*, Imperial College Press, London UK, 2005.

[17] G. A. Bunin, *On the equivalence between the modifier-adaptation and trust-region frameworks*, Comp. Chem. Engng., 71 (2014), pp. 154 – 157.

[18] G. A. Bunin, G. François, and D. Bonvin, *From discrete measurements to bounded gradient estimates: A look at some regularizing structures*, Ind. Eng. Chem. Res., 52 (2013), pp. 12500–12513.

[19] ——, *Sufficient conditions for feasibility and optimality of real-time optimization schemes - i. Theoretical foundations*. arXiv:1308.2620, 2013.

[20] G. A. Bunin, Z. Wuillemin, G. François, A. Nakajo, L. Tsikonis, and D. Bonvin, *Experimental real-time optimization of a solid oxide fuel cell stack via constraint adaptation*, Energy, 39 (2012), pp. 54–62.

[21] B. Chachuat, A. Marchetti, and D. Bonvin, *Process optimization via constraints adaptation*, J. Process Contr., 18 (2008), pp. 244–257.

[22] B. Chachuat, B. Srinivasan, and D. Bonvin, *Adaptation strategies for real-time optimization*, Comp. Chem. Eng., 33 (2009), pp. 1557–1567.

[23] S. Chan, K. Khor, and Z. Xia, *A complete polarization model of a solid oxide fuel cell and its sensitivity to the change of cell component thickness*, J. Power Sources, 93 (2001), pp. 130 – 140.

[24] C. Y. Chen and B. Joseph, *On-line optimization using a two-phase approach: An application study*, Ind. Eng. Chem. Res., 26 (1987), pp. 1924–1930.

[25] R. Confalonieri, G. Bellocchi, S. Bregaglio, M. Donatelli, and M. Acutis, *Comparison of sensitivity analysis techniques: A case study with the rice model warm*, Ecological Modelling, 221 (2010), pp. 1897 – 1906.

[26] A. Conn, N. Gould, and P. Toint, *Trust Region Methods*, vol. 1, MOS-SIAM Series on Optimization, Philadelphia, PA, USA, 2000.

[27] A. Conn, K. Scheinberg, and L. Vicente, *Introduction to Derivative-Free Optimization*, MOS-SIAM Series on Optimization, Philadelphia, PA, USA, 2009.

[28] P. G. Constantine, *Active Subspaces: Emerging Ideas for Dimension Reduction in Parameter Studies*, SIAM, Philadelphia, PA, USA, 2015.

[29] P. G. Constantine, E. Dow, and Q. Wang, *Active subspace methods in theory and practice: Applications to kriging surfaces*, SIAM Journal on Scientific Computing, 36 (2014), pp. A1500–A1524.

[30] R. D. Cook, *Regression Graphics: Ideas for Studying Regressions through Graphics*, vol. 482, Wiley Series in Probability & Statistics, New York: John Wiley & Sons, 2009.

[31] S. Costello, *Real-Time Optimization via Directional Modifier Adaptation, with Application to Kite Control*, EPFL thesis, no 6571, École Polytechnique Fédérale de Lausanne, 2015.

[32] S. Costello, G. François, and D. Bonvin, *Directional real-time optimization applied to a kite-control simulation benchmark*, in 2015 European Control Conference (ECC), July 2015, pp. 1594–1601.

[33] ——, *A directional modifier-adaptation algorithm for real-time optimization*, J. Process Contr., 39 (2016), pp. 64–76.

[34] ——, *Real-time optimizing control of an experimental crosswind power kite*, IEEE Trans. Contr. Sys. Tech., PP (2017), pp. 1–16.

[35] S. Costello, G. François, B. Srinivasan, and D. Bonvin, *Modifier adaptation for run-to-run optimization of transient processes*, in Proceedings of the 18th IFAC World Congress, Milan, Italy, 28 August–2 September 2011.

[36] T. Crestaux, O. Le Maître, and J.-M. Martinez, *Polynomial chaos expansion for sensitivity analysis*, Reliability Engineering & System Safety, 94 (2009), pp. 1161–1172.

[37] C. R. Cutler and R. T. Perry, *Real time optimization with multivariable control is required to maximize profits*, Comp. Chem. Engng., 7 (1983), pp. 663–667.

[38] T. de Avila Ferreira, G. François, A. G. Marchetti, and D. Bonvin, *Use of transient measurements for static real-time optimization*, IFAC-PapersOnLine, 50 (2017), pp. 5737 – 5742. 20th IFAC World Congress.

[39] S. Diwale, T. Faulwasser, and C. N. Jones, *Model predictive path-following control for airborne wind energy systems*, IFAC-PapersOnLine, 50 (2017), pp. 13270 – 13275.

[40] M. Engelhart, D. Lebiedz, and S. Sager, *Optimal control for selected cancer chemotherapy ode models: A view on the potential of optimal schedules and choice of objective function*, Mathematical Biosciences, 229 (2011), pp. 123 – 134.

[41] M. Erhard and H. Strauch, *Control of towing kites for seagoing vessels*, IEEE Trans. Contr. Syst. Tech., 21 (2013), pp. 1629–1640.

[42] T. Faulwasser and D. Bonvin, *On the use of second-order modifiers for real-time optimization*, IFAC Proceedings Volumes, 47 (2014), pp. 7622 – 7628. 19th IFAC World Congress.

[43] J. F. Forbes and T. E. Marlin, *Model accuracy for economic optimizing controllers: The bias update case*, Ind. Eng. Chem. Res., 33 (1994), pp. 1919–1929.

[44] ——, *Model adequacy requirements for optimizing plant operations*, Comp. Chem. Engng., 18 (1994), pp. 497–510.

[45] ——, *Design cost: A systematic approach to technology selection for model-based real-time optimization systems*, Comp. Chem. Engng., 20 (1996), pp. 717–734.

[46] G. François, *Measurement-based run-to-run optimization of batch processes application to industrial acrylamide copolymerization*, EPFL thesis, no 3128, École Polytechnique Fédérale de Lausanne, 2005.

[47] G. François and D. Bonvin, *Measurement-based real-time optimization of chemical processes*, in Control and Optimisation of Process Systems, S. Pushpavanam, ed., vol. 43, Academic Press, Oxford, 2013, pp. 1–50.

[48] G. François and D. Bonvin, *Use of convex model approximations for real-time optimization via modifier adaptation*, Ind. Eng. Chem. Res., 52 (2013), pp. 11614–11625.

[49] G. François, B. Srinivasan, and D. Bonvin, *Use of measurements for enforcing the necessary conditions of optimality in the presence of constraints and uncertainty*, J. Process Contr., 15 (2005), pp. 701–712.

[50] G. François, B. Srinivasan, and D. Bonvin, *Equivalence between neighboring-extremal control and self-optimizing control for the steady-state optimization of dynamical systems*, Ind. Eng. Chem. Res., 53 (2014), pp. 7470–7478.

[51] G. François, B. Srinivasan, D. Bonvin, and D. Hunkeler, *Measurement-based optimization of an emulsion polymerization process*, Polymer Reaction Engineering: Modelling, Optimization and Control, Keynote lecture (2003), pp. 16–19.

[52] G. François, B. Srinivasan, and D. Bonvin, *Comparison of six implicit real-time optimization schemes*, Journal Européen des Systèmes Automatisés, 46 (2012), pp. 291–305.

[53] F. Fritz, *Application of an Automated Kite System for Ship Propulsion and Power Generation*, Springer Berlin Heidelberg, 2013, pp. 359–372.

[54] F. Galvanin, C. C. Ballan, M. Barolo, and F. Bezzo, *A general model-based design of experiments approach to achieve practical identifiability of pharmacokinetic and pharmacodynamic models*, J. Pharmacokinetics and Pharmacodynamics, 40 (2013), pp. 451–467.

[55] F. Galvanin, R. Marchesini, M. Barolo, F. Bezzo, and M. Fidaleo, *Optimal design of experiments for parameter identification in electrodialysis models*, Chemical Engineering Research and Design, 105 (2016), pp. 107 – 119.

[56] W. Gao and S. Engell, *Iterative set-point optimization of batch chromatography*, Comput. Chem. Engng., 29 (2005), pp. 1401–1409.

[57] W. Gao, S. Wenzel, and S. Engell, *Comparison of modifier adaptation schemes in real-time optimization*, IFAC-PapersOnLine, 48 (2015), pp. 182 – 187. 9th IFAC Symposium on Advanced Control of Chemical Processes ADCHEM 2015.

[58] W. Gao, S. Wenzel, and S. Engell, *A reliable modifier-adaptation strategy for real-time optimization*, Comp. Chem. Eng., 91 (2016), pp. 318–328. 12th Int. Symp. on Process Systems Engineering & 25th European Symp. of Computer Aided Process Engineering, Copenhagen, Denmark.

[59] P. E. Gill, W. Murray, and M. H. Wright, *Practical Optimization*, London: Academic Press, UK, 2003.

[60] J. Golbert and D. R. Lewin, *Model-based control of fuel cells (2): Optimal efficiency*, J. Power Sources, 173 (2007), pp. 298 – 309.

[61] S. Gros, B. Srinivasan, B. Chachuat, and D. Bonvin, *Neighbouring-extremal control for singular dynamic optimisation problems. part i: Single-input systems*, Int. J. Control, 82 (2009), pp. 1099–1112.

[62] M. Guay and T. Zhang, *Adaptive extremum seeking control of nonlinear dynamic systems with parametric uncertainties*, Automatica, 39 (2003), pp. 1283–1293.

[63] B. Iooss and P. Lemaître, *A review on global sensitivity analysis methods*, in Uncertainty management in simulation-optimization of complex systems, Springer, 2015, pp. 101–122.

[64] M. Itik, M. U. Salamci, and S. P. Banks, *Optimal control of drug therapy in cancer treatment*, Nonlinear Analysis: Theory, Methods & Applications, 71 (2009), pp. e1473 – e1486.

[65] S.-S. Jang, B. Joseph, and H. Mukai, *On-line optimization of constrained multivariable chemical processes*, AIChE J., 33 (1987), pp. 26–35.

[66] A. T. Jarullah, I. M. Mujtaba, and A. S. Wood, *Kinetic parameter estimation and simulation of trickle-bed reactor for hydrodesulfurization of crude oil*, Chemical Engineering Science, 66 (2011), pp. 859 – 871.

[67] J. Jäschke, Y. Cao, and V. Kariwala, *Self-optimizing control–a survey*, Annual Reviews in Control, 43 (2017), pp. 199–223.

[68] J. Jäschke, M. Fikar, and S. Skogestad, *Self-optimizing invariants in dynamic optimization*, in 2011 50th IEEE Conference on Decision and Control and European Control Conference, Dec 2011, pp. 7753–7758.

[69] J. Jäschke and S. Skogestad, *NCO tracking and self-optimizing control in the context of real-time optimization*, J. Proc. Contr., 21 (2011), pp. 1407–1416.

[70] M. Jonin, M. Singhal, S. Diwale, C. N. Jones, and D. Bonvin, *Active directional modifier adaptation with trust region – application to energy-harvesting kites*, in Proc. 17th annual European Control Conference, Limassol, Cyprus, 2018.

[71] A. Karimi and C. Kammer, *A data-driven approach to robust control of multivariable systems by convex optimization*, Automatica, 85 (2017), pp. 227–233.

[72] S. Katare, A. Bhan, J. M. Caruthers, W. N. Delgass, and V. Venkatasubramanian, *A hybrid genetic algorithm for efficient parameter estimation of large kinetic models*, Comp. Chem. Engng., 28 (2004), pp. 2569 – 2581.

[73] M. Krstic and H.-H. Wang, *Stability of extremum seeking feedback for general nonlinear dynamic systems*, Automatica, 36 (2000), pp. 595 – 601.

[74] S. Kucherenko, M. Rodriguez-Fernandez, C. Pantelides, and N. Shah, *Monte Carlo evaluation of derivative-based global sensitivity measures*, Reliability Engineering & System Safety, 94 (2009), pp. 1135 – 1148. Special Issue on Sensitivity Analysis.

[75] I. Landau, R. Lozano, M. M'Saad, and A. Karimi, *Adaptive Control Algorithms, Analysis and Applications*, Communications and Control Engineering, London: Springer, 2011.

[76] D. Larrain, J. V. herle, and D. Favrat, *Simulation of SOFC stack and repeat elements including interconnect degradation and anode reoxidation risk*, J. Power Sources, 161 (2006), pp. 392 – 403.

[77] A. Lewis, R. C. Smith, and B. Williams, *Gradient free active subspace construction using morris screening elementary effects*, Computers & Mathematics with Applications, 72 (2016), pp. 1603–1615.

[78] F. S. Liporace, M. V. Gomes, A. C. Katata, A. C. Zanin, L. F. Moro, and C. R. PorfÃrio, *Petrobras experience implementing real time optimization*, in 10th International Symposium on Process Systems Engineering: Part A, vol. 27 of Computer Aided Chemical Engineering, Elsevier, 2009, pp. 1245 – 1250.

[79] J. S. Mandur and H. M. Budman, *Robust algorithms for simultaneous model identification and optimization in the presence of model-plant mismatch*, Ind. Eng. Chem. Res., 54 (2015), pp. 9382–9393.

[80] ——, *Simultaneous model identification and optimization in presence of model-plant mismatch*, Chem. Engng. Sci., 129 (2015), pp. 106–115.

[81] M. Mansour and J. Ellis, *Comparison of methods for estimating real process derivatives in on-line optimization*, App. Math. Modelling, 27 (2003), pp. 275–291.

[82] A. Marchetti, *Modifier-Adaptation Methodology for Real-Time Optimization*, PhD thesis, # 4449, Ecole Polytechnique Fédérale de Lausanne, 2009.

[83] A. Marchetti, B. Chachuat, and D. Bonvin, *Modifier-adaptation methodology for real-time optimization*, Ind. Eng. Chem. Res., 48 (2009), pp. 6022–6033.

[84] ——, *A dual modifier-adaptation approach for real-time optimization*, J. Process Contr., 20 (2010), pp. 1027–1037.

[85] A. Marchetti, T. Faulwasser, and D. Bonvin, *A feasible-side globally convergent modifier-adaptation algorithm*, J. Process Contr., 54 (2017), pp. 38–46.

[86] A. Marchetti, G. François, T. Faulwasser, and D. Bonvin, *Modifier adaptation for real-time optimization – Methods and applications*, Processes, 4 (2016), pp. 1–35.

[87] A. Marchetti, A. Gopalakrishnan, B. Chachuat, D. Bonvin, L. Tsikonis, A. Nakajo, Z. Wuillemin, and J. Van herle, *Robust real-time optimization of a solid oxide fuel cell stack*, J. Fuel Cell Sci. Technol., 8 (2011).

[88] A. Marchetti, M. Singhal, T. Faulwasser, and D. Bonvin, *Modifier adaptation with guaranteed feasibility in the presence of gradient uncertainty*, Comp. Chem. Engng., 98 (2017), pp. 61 – 69.

[89] T. E. Marlin and A. N. Hrymak, *Real-time operations optimization of continuous processes*, in AICHE Symposium Series - CPC-V, vol. 93, 1997, pp. 156–164.

[90] P. Milosavljevic, R. Schneider, A. Cortinovis, T. Faulwasser, and D. Bonvin, *A distributed feasible-side convergent modifier-adaptation scheme for interconnected systems, with application to gas-compressor stations*, Comp. Chem. Engng., 115 (2018), pp. 474 – 486.

[91] M. D. Morris, *Factorial sampling plans for preliminary computational experiments*, Technometrics, 33 (1991), pp. 161–174.

[92] J. Murray, *Optimal control for a cancer chemotheraphy problem with general growth and loss functions*, Mathematical Biosciences, 98 (1990), pp. 273 – 287.

[93] D. Navia, L. Briceño, G. Gutiérrez, and C. De Prada, *Modifier-adaptation methodology for real-time optimization reformulated as a nested optimization problem*, Ind. Eng. Chem. Res., 54 (2015), pp. 12054–12071.

[94] D. Navia, G. Gloria, and C. De Prada, *Nested modifier-adaptation for rto in the otto williams reactor*, IFAC Proc. Vol., 46 (2013), pp. 123 – 128. 10th IFAC International Symposium on Dynamics and Control of Process Systems.

[95] D. Navia, D. Villegas, I. Cornejo, and C. de Prada, *Real-time optimization for a laboratory-scale flotation column*, Comp. Chem. Engng., 86 (2016), pp. 62 – 74.

[96] R. Oeuvray and M. Bierlaire, *Boosters: A derivative-free algorithm based on radial basis functions*, Int. J. Modelling and Simulation, 29 (2009), pp. 26–36.

[97] A. B. Owen, *Variance components and generalized sobol'indices*, SIAM/ASA J. Uncertainty Quantification, 1 (2013), pp. 19–41.

[98] J. Park and I. W. Sandberg, *Universal approximation using radial-basis-function networks*, Neural computation, 3 (1991), pp. 246–257.

[99] Z. Qian, C. C. Seepersad, V. R. Joseph, J. K. Allen, and C. J. Wu, *Building surrogate models based on detailed and approximate simulations*, J. Mechanical Design, 128 (2006), pp. 668–677.

[100] F. Ramadhani, M. Hussain, H. Mokhlis, and S. Hajimolana, *Optimization strategies for solid oxide fuel cell (sofc) application: A literature survey*, Renewable and Sustainable Energy Reviews, 76 (2017), pp. 460 – 484.

[101] P. D. Roberts, *An algorithm for steady-state system optimization and parameter estimation*, Int. J. Systems Science, 10 (1979), pp. 719–734.

[102] ——, *Coping with model-reality differences in industrial process optimisation–a review of integrated system optimisation and parameter estimation (ISOPE)*, Computers in Industry, 26 (1995), pp. 281 – 290. Computer Integrated Manufacturing and Industrial Automation.

[103] P. D. Roberts and T. W. C. Williams, *On an algorithm for combined system optimisation and parameter estimation*, Automatica, 17 (1981), pp. 199–209.

[104] E. A. Rodger and B. Chachuat, *Design methodology of modifier adaptation for on-line optimization of uncertain processes*, IFAC Proceedings Volumes, 44 (2011), pp. 4113–4118.

[105] T. Rodríguez-Blanco, D. Sarabia, D. Navia, and C. de Prada, *Efficient nested modifier adaptation for rto using lagrangian functions*, in 27th European Symposium on Computer Aided Process Engineering, vol. 40 of Computer Aided Chemical Engineering, Elsevier, 2017, pp. 1723 – 1728.

[106] J. Ruess, F. Parise, A. Milias-Argeitis, M. Khammash, and J. Lygeros, *Iterative experiment design guides the characterization of a light-inducible gene expression circuit*, Proceedings of the National Academy of Sciences, 112 (2015), pp. 8148–8153.

[107] W. J. Rugh and J. S. Shamma, *Research on gain scheduling*, Automatica, 36 (2000), pp. 1401 – 1425.

[108] R. Ruiterkamp and S. Sieberling, *Description and Preliminary Test Results of a Six Degrees of Freedom Rigid Wing Pumping System*, Springer Berlin Heidelberg, 2013, pp. 443–458.

[109] D. Ruppen, D. Bonvin, and D. W. T. Rippin, *Implementation of adaptive optimal operation for a semi-batch reaction system*, Comp. Chem. Engng., 22 (1997), pp. 185–199.

[110] T. M. Russi, *Uncertainty Quantification with Experimental Data and Complex System Models*, PhD thesis, University of California, Berkeley, 2010.

[111] N. P. Salau, J. O. Trierweiler, and A. R. Secchi, *Observability analysis and model formulation for nonlinear state estimation*, App. Math. Modelling, 38 (2014), pp. 5407–5420.

[112] A. Saltelli, M. Ratto, T. Andres, F. Campolongo, J. Cariboni, D. Gatelli, M. Saisana, and S. Tarantola, *Global Sensitivity Analysis: The Primer*, New York: John Wiley & Sons, 2008.

[113] A. Saltelli, M. Ratto, S. Tarantola, and F. Campolongo, *Sensitivity analysis practices: Strategies for model-based inference*, Reliability Engineering & System Safety, 91 (2006), pp. 1109 – 1125. The Fourth International Conference on Sensitivity Analysis of Model Output (SAMO 2004).

[114] R. Schneider, P. Milosavljevic, and D. Bonvin, *Distributed modifier-adaptation schemes for the real-time optimisation of uncertain interconnected systems*, Int. J. Control, 0 (2017), pp. 1–14.

[115] R. Singh, M. Sen, M. Ierapetritou, and R. Ramachandran, *Integrated moving horizon-based dynamic real-time optimization and hybrid mpc-pid control of a direct compaction continuous tablet manufacturing process*, J. Pharmaceutical Innovation, 10 (2015), pp. 233–253.

[116] M. Singhal, T. Faulwasser, and D. Bonvin, *On handling cost gradient uncertainty in real-time optimization*, IFAC-PapersOnLine, 48 (2015), pp. 176–181.

[117] M. Singhal, T. Faulwasser, and D. Bonvin, *Adaptation strategies for tracking constraints under plant-model mismatch*, in Proc. 10th IFAC Symposium on Advanced Control of Chemical Processes, Shenyang, China, 2018.

[118] M. Singhal, A. Marchetti, T. Faulwasser, and D. Bonvin, *Improved directional derivatives for modifier-adaptation schemes*, IFAC-PapersOnLine, 50 (2017), pp. 5718 – 5723. 20th IFAC World Congress.

[119] M. Singhal, A. Marchetti, T. Faulwasser, and D. Bonvin, *Active directional modifier adaptation for real-time optimization*, Comp. Chem. Engng., 115 (2018), pp. 246–261.

[120] M. Singhal, A. G. Marchetti, T. Faulwasser, and D. Bonvin, *Real-time optimization based on adaptation of surrogate models*, IFAC-PapersOnLine, 49 (2016), pp. 412 – 417. 11th IFAC Symposium on Dynamics and Control of Process SystemsIncluding Biosystems DYCOPS-CAB 2016.

[121] S. Skogestad, *Plantwide control: The search for the self-optimizing control structure*, J. Proc. Contr., 10 (2000), pp. 487–507.

[122] S. Skogestad and I. Postlethwaite, *Multivariable feedback control: analysis and design*, vol. 2, New York: John Wiley & Sons, 2007.

[123] R. C. Smith, *Uncertainty Quantification: Theory, Implementation, and Applications*, SIAM Computational Science & Engineering Series: Philadelphia, USA, 2014.

[124] I. M. Sobol, *Global sensitivity indices for nonlinear mathematical models and their monte carlo estimates*, Mathematics and Computers in Simulation, 55 (2001), pp. 271 – 280. The Second IMACS Seminar on Monte Carlo Methods.

[125] B. Srinivasan and D. Bonvin, *Real-time optimization of batch processes by tracking the necessary conditions of optimality*, Ind. Eng. Chem. Res., 46 (2007), pp. 492–504.

[126] B. Srinivasan, D. Bonvin, E. Visser, and S. Palanki, *Dynamic optimization of batch processes: II. Role of measurements in handling uncertainty*, Comp. Chem. Engng., 27 (2003), pp. 27 – 44.

[127] B. Srinivasan, S. Palanki, and D. Bonvin, *Dynamic optimization of batch processes: I. Characterization of the nominal solution*, Comp. Chem. Engng., 27 (2003), pp. 1 – 26.

[128] M. L. Stein, *Interpolation of spatial data: some theory for kriging*, New York: Springer Verlag, 1999.

[129] G. Tao, *Multivariable adaptive control: A survey*, Automatica, 50 (2014), pp. 2737 – 2764.

[130] T. Williams and R. Otto, *A generalized chemical processing model for the investigation of computer control*, Trans. American Institute of Electrical Engineers, Part I: Communication and Electronics, 79 (1960), pp. 458–473.

[131] Z. Wuillemin, N. Autissier, A. Nakajo, M. Luong, J. V. herle, and D. Favrat, *Modeling and study of the influence of sealing on a solid oxide fuel cell*, J. Fuel Cell Sci. Tech., 5(1), 011016 (2008).

[132] L. Würth, R. Hannemann, and W. Marquardt, *Neighboring-extremal updates for nonlinear model-predictive control and dynamic real-time optimization*, J. Process Contr., 19 (2009), pp. 1277–1288.

[133] W. S. Yip and T. E. Marlin, *The effect of model fidelity on real-time optimization performance*, Comp. Chem. Engng., 28 (2004), pp. 267 – 280. Escape 12.

[134] A. U. Zgraggen, L. Fagiano, and M. Morari, *Real-time optimization and adaptation of the crosswind flight of tethered wings for airborne wind energy*, IEEE Trans. Contr. Syst. Tech., 23 (2015), pp. 434–448.

[135] J. Zhang, Z. zhong Mao, R. da Jia, and D. kuo He, *Real time optimization based on a serial hybrid model for gold cyanidation leaching process*, Minerals Engineering, 70 (2015), pp. 250 – 263.

[136] X. Zhang, S. Chan, H. Ho, J. Li, G. Li, and Z. Feng, *Nonlinear model predictive control based on the moving horizon state estimation for the solid oxide fuel cell*, Int. J. Hydrogen Energy, 33 (2008), pp. 2355 – 2366.

Martand Singhal

✉ martand.singhal@gmail.com

Research Experience

2013–to date **Doctoral assistant**, *Ecole Polytechnique Fédérale de Lausanne*, Switzerland

• Developed a framework that examines a given process model's ability to independently enforce the plant's necessary conditions of optimality by exploiting the cutting-edge tools of uncertainty quantification.

• Led an industrial project on "Modeling and optimization of fuel cell systems for automotive applications".

• Supervised 9 master students on the subjects of process modeling and optimization.

• Worked as teaching assistant for "control systems", "probabilities and statistics", "systems dynamics" and "systems identification" courses.

Education

2013–2018 **PhD in process systems engineering**
Ecole Polytechnique Fédérale de Lausanne, Lausanne, Switzerland
Thesis title: On enforcing the necessary conditions of optimality under plant-model mismatch - what to measure and what to adapt?
Thesis directors: Prof. Dominique Bonvin and Dr. Timm Faulwasser.

2009–2011 **Master in systems and control engineering, CPI (9.25/10)**
Indian Institute of Technology Bombay, Mumbai, India

• Granted full scholarship on achieving the rank of 116 out of 4000+ chemical engineers appeared for the prestigious all-India level GATE exam, 2009.

2005–2009 **Bachelor in chemical engineering, CPI (71.23/100)**
University School of Chemical Technology, G.G.S.I.P University, Delhi, India

Publications

M. Singhal, A. Marchetti, T. Faulwasser, and D. Bonvin. Active directional modifier adaptation for real-time optimization. *Computers & Chemical Engineering*, 115:246–261, 2018.

M. Singhal, T. Faulwasser, and D. Bonvin. Adaptation strategies for tracking constraints under plant-model mismatch. In *Proc. 10th IFAC Symposium on Advanced Control of Chemical Processes*, Shenyang, China, 2018.

M. Jonin, M. Singhal, S. Diwale, C. N. Jones, and D. Bonvin. Active directional modifier adaptation with trust region – application to energy-harvesting kites. In *Proc. 17th annual European Control Conference*, Limassol, Cyprus, 2018.

A. Ahmad, M. Singhal, W. Gao, D. Bonvin, and S. Engell. Enforcing model adequacy in real-time optimization via dedicated parameter adaptation. In *Proc. 10th IFAC Symposium on Advanced Control of Chemical Processes*, Shenyang, China, 2018.

M. Singhal, A. Marchetti, T. Faulwasser, and D. Bonvin. Improved directional derivatives for modifier-adaptation schemes. *IFAC-PapersOnLine*, 50(1):5718 – 5723, 2017. 20th IFAC World Congress.

A. Marchetti, M. Singhal, T. Faulwasser, and D. Bonvin. Modifier adaptation with guaranteed feasibility in the presence of gradient uncertainty. *Computers & Chemical Engineering*, 98:61 – 69, 2017.

M. Singhal, A. Marchetti, T. Faulwasser, and D. Bonvin. Real-time optimization based on adaptation of surrogate models. *IFAC-PapersOnLine*, 49(7):412 – 417, 2016. 11th IFAC Symposium on Dynamics and Control of Process Systems Including Biosystems DYCOPS-CAB.

M. Singhal, T. Faulwasser, and D. Bonvin. On handling cost gradient uncertainty in real-time optimization. *IFAC-PapersOnLine*, 48(8):176 – 181, 2015. 9th IFAC Symposium on Advanced Control of Chemical Processes ADCHEM 2015.

M. Singhal and R. D. Gudi. Optimal batch process regulation using self-optimizing control, nco tracking. *IFAC Proceedings Volumes*, 45(15):81 – 86, 2012. 8th IFAC Symposium on Advanced Control of Chemical Processes.