

Modifier-Adaptation Methodology for Real-Time Optimization

THÈSE N° 4449 (2009)

PRÉSENTÉE LE 9 JUILLET 2009

À LA FACULTÉ ET TECHNIQUES DE L'INGÉNIEUR

LABORATOIRE D'AUTOMATIQUE

PROGRAMME DOCTORAL EN INFORMATIQUE, COMMUNICATIONS ET INFORMATION

ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

POUR L'OBTENTION DU GRADE DE DOCTEUR ÈS SCIENCES

PAR

Alejandro Gabriel MARCHETTI

acceptée sur proposition du jury:

Prof. R. Longchamp, président du jury
Prof. D. Bonvin, Dr B. Chachuat, directeurs de thèse
Dr F. Maréchal, rapporteur
Dr M. Mercangöz, rapporteur
Prof. B. Ogunnaïke, rapporteur



ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE

Suisse
2009

a mi familia

Acknowledgements

This thesis would not have been possible without the help and support of numerous people. I would first like to express my deepest gratitude to my adviser: Professor Dominique Bonvin, for giving me the opportunity to join the Laboratoire d'Automatique (LA), initially as a scholarship holder of the Swiss Confederation, and subsequently as a PhD student. Professor Bonvin transmitted to me his vision about the field, and gave me the freedom and confidence that allowed me to explore different things and find the subject of my thesis. He closely followed my work, sharing with me his expertise and providing research insight.

I am very thankful to my co-adviser, Dr. Benoît Chachuat, who has largely contributed to improve the quality of this thesis. I have learned a lot by working with him.

I would like to thank the members of my thesis jury: Prof. Roland Longchamp, Prof. Dominique Bonvin, Dr. Benoît Chachuat, MER François Maréchal, Dr. Mehmet Mercangöz and Prof. Babatunde Ogunnaike.

I wish to express my appreciation to Dr. Bala Srinivasan for supervising my work during my first years at the LA as a scholarship holder, as well as my work during my initial years as a doctoral student. He is the person who guided my first steps into the exciting world of research.

I am profoundly thankful to the former secretary of the LA, Marie Claire. The help she provided during the first two years when I was a scholarship holder was invaluable to me. I am equally thankful to all the secretary team: Ruth, Homeira, Francine and Sol. They take well care of all administrative issues, making work be more enjoyable.

I would like to thank Philippe Cuanillon who has kept my computer working at the optimal operating conditions during all these years.

I would like to thank my colleagues at LA for their friendship and for contributing to the pleasant atmosphere at LA. I would like to thank all the friends I met in Switzerland, and the friends I left back in Argentina, for all the moments we shared together, for their support and their encouragement.

Finally, I owe my deepest gratitude to my family for all their unconditional love and support during all these years I lived abroad. My mother, Olga, my father, Jacinto, my sister, María Celia, and my brother, Pablo, to whom I dedicate this thesis.

Abstract

The process industries are characterized by a large number of continuously operating plants, for which optimal operation is of economic importance. However, optimal operation is particularly difficult to achieve when the process model used in the optimization is inaccurate or in the presence of process disturbances.

In highly automated plants, optimal operation is typically addressed by a decision hierarchy involving several levels that include plant scheduling, real-time optimization (RTO), and process control. At the RTO level, medium-term decisions are made by considering economic objectives explicitly. This step typically relies on an optimizer that determines the optimal steady-state operating point under slowly changing conditions such as catalyst decay or changes in raw material quality. This optimal operating point is characterized by setpoints that are passed to lower-level controllers. Model-based RTO typically involves nonlinear first-principles models that describe the steady-state behavior of the plant.

Since accurate models are rarely available in industrial applications, RTO typically proceeds using an iterative two-step approach, namely a parameter estimation step followed by an optimization step. The idea is to repeatedly estimate selected uncertain model parameters and use the updated model to generate new inputs via optimization. This way, the model is expected to yield a better description of the plant at its current operating point. The classical two-step approach works well provided that (i) there is little structural plant-model mismatch, and (ii) the changing operating conditions provide sufficient excitation for estimating the uncertain model parameters. Unfortunately, such con-

ditions are rarely met in practice and, in the presence of plant-model mismatch, the algorithm might not converge to the plant optimum, or worse, to a feasible operating point. As far as feasibility is concerned, the updated model should be able to match the plant constraints.

Alternatively, feasibility can be enforced without requiring the solution of a parameter estimation problem by adding plant-model bias terms to the model outputs. These biases are obtained by subtracting the model outputs from the measured plant outputs. A bias-update scheme, where the bias terms are used to modify the constraints in the steady-state optimization problem, has been used in industry. However, the analysis of this scheme has received little attention in the research community. In the context of this thesis, such an RTO scheme is referred to as *constraint adaptation*. The constraint-adaptation scheme is studied, and its local convergence properties are analyzed.

Constraint adaptation guarantees reaching a feasible operating point upon convergence. However, the constraints might be violated during the iterations of the algorithm, even when starting the adaptation from within the feasible region. Constraint violations can be avoided by controlling the constraints in the optimization problem, which is done at the process control level by means of model predictive control (MPC). The approach for integrating constraint adaptation with MPC described in this thesis places high emphasis on how constraints are handled. An alternative constraint-adaptation scheme is proposed, which permits one to move the constraint setpoints gradually in the constraint controller. The constraint-adaptation scheme, with and without the constraint controller, is illustrated in simulation through the real-time optimization of a fuel-cell system.

It is desirable for a RTO scheme to achieve both feasibility and optimality. Optimality can be achieved if the underlying process model is able to predict not only the constraint values of the plant, but also the gradients of the cost and constraint functions. In the presence of structural plant-model mismatch, this typically requires the use of experimental plant gradient information. Methods integrating parameter estimation with a modified optimization problem that uses plant gradient information have been studied in the literature. The approach studied in this thesis, denoted *modifier adaptation*, does not require parameter estimation. In addition to the modifiers used in constraint adaptation, gradient-modifier terms based on the difference between the estimated and predicted gradient values are added to the cost and constraint functions in the optimization problem. With this, a point

that satisfies the first-order necessary conditions of optimality for the plant is obtained upon convergence. The modifier-adaptation scheme is analyzed in terms of model adequacy and local convergence conditions. Different filtering strategies are discussed. The constraint-adaptation and modifier-adaptation RTO approaches are illustrated experimentally on a three-tank system.

Finite-difference techniques can be used to estimate experimental gradients. The *dual modifier-adaptation* approach studied in this thesis drives the process towards optimality, while paying attention to the accuracy of the estimated gradients. The gradients are estimated from the successive operating points generated by the optimization algorithm. A novel upper bound on the gradient estimation error is developed, which is used as a constraint for locating the next operating point.

Keywords: Real-time optimization, constraint adaptation, constraint control, modifier adaptation, estimation of experimental gradients.

Version abrégée

Les procédés industriels sont typiquement composés d'un grand nombre d'installations opérant de façon continue, pour lesquelles l'opération optimale revêt une grande importance économique. Cependant, il est particulièrement difficile de déterminer les conditions optimales de fonctionnement lorsque l'on dispose de modèles de procédés imprécis ou en présence de perturbations. Dans des usines hautement automatisées, l'opération optimale est généralement réalisée par une hiérarchie de décisions, sur plusieurs niveaux, comme la planification des opérations, l'optimisation en temps réel, et la commande des procédés.

Au niveau de l'optimisation en temps réel, des décisions à moyen terme sont prises en considérant les objectifs économiques de façon explicite. Ce niveau se base sur un optimiseur qui détermine le point de fonctionnement optimal, à l'état stationnaire, pour des perturbations à dynamiques lentes, telles la dégradation d'un catalyseur ou une variation de la qualité des matières premières. En pratique, il s'agit de déterminer les consignes qui seront assignées à des régulateurs à un niveau de hiérarchie plus bas, c'est-à-dire au niveau de la commande de procédés. L'optimisation en temps réel basée sur le modèle implique typiquement l'utilisation de modèles de connaissance qui décrivent le comportement du procédé à l'état stationnaire.

Comme des modèles précis sont rarement disponibles pour les applications industrielles, l'implantation classique de l'optimisation en temps réel se fait de deux étapes, à savoir une étape d'estimation de paramètres, suivie d'une étape d'optimisation. L'idée est d'utiliser les mesures disponibles pour identifier la valeur d'un certain nombre de

paramètres incertains du modèle, choisis a priori. Suite à la mise à jour du modèle du procédé, une optimisation basée sur le modèle est réalisée pour déterminer un nouveau point de fonctionnement pour le procédé réel. Enfin, cette séquence d'étapes est répétée aussi longtemps que le comportement du procédé réel diffère de celui du modèle, et le comportement du modèle est ainsi sensé se rapprocher itérativement du comportement du procédé. Cette approche donne de bons résultats dans le cas où (i) les erreurs structurelles du modèle sont faibles, et (ii) la variation des points de fonctionnement fournit une excitation suffisante pour estimer les paramètres incertains du modèle.

Malheureusement, de telles conditions sont rarement réunies dans la pratique et, en présence d'erreurs structurelles dans le modèle, cette approche classique peut ne pas converger à l'optimum du procédé réel, ou pire, peut converger vers un point infaisable, c'est-à-dire violant les contraintes du procédé réel. Dans ce contexte, la seule garantie de faisabilité à la convergence pour l'approche classique réside dans la capacité du modèle mis à jour de pouvoir estimer avec précision les contraintes du procédé réel.

Alternativement, il est possible de garantir la faisabilité sans recourir à la résolution d'un problème d'estimation paramétrique, par l'ajout des termes de correction aux sorties du modèle. Ces termes correctifs sont obtenus en soustrayant les sorties du modèle aux sorties mesurées du procédé. Une approche de ce genre, pour laquelle des termes de correction sont employés pour modifier les contraintes du problème d'optimisation d'état stationnaire, a déjà été employée avec succès dans l'industrie. Cependant, rares sont les travaux traitant de façon théorique de l'analyse de cette approche dans la littérature. Cette thèse développe cette approche d'optimisation en temps réel, nommée *adaptation des contraintes*, et étudie ses propriétés locales de convergence.

L'analyse de convergence montre que l'adaptation de contraintes garantit, sous réserve de convergence, l'atteinte d'un point de fonctionnement faisable. Cependant, sans autres précautions, les contraintes peuvent être violées pendant les itérations de l'algorithme, même lorsque l'adaptation est initialisée dans le domaine de faisabilité. Ces violations de contraintes peuvent être évitées à un niveau hiérarchique plus bas, celui de la commande de procédés, en contrôlant directement les contraintes du problème d'optimisation, ce qui est réalisé par commande prédictive. Dans cette thèse, une nouvelle approche pour combiner l'adaptation des contraintes et la commande prédictive est

décrite. L'accent est mis sur la façon dont les contraintes sont manipulées, et la méthode proposée permet de déplacer graduellement les consignes des contraintes dans le régulateur de contraintes. L'approche, avec et sans le régulateur de contraintes, est illustrée par l'optimisation en temps réel d'un système de pile à combustible simulé.

Si la faisabilité est nécessaire, il est souhaitable que l'approche d'optimisation assure en outre l'optimalité. L'optimalité peut être garantie si le modèle du processus peut prédire non seulement les valeurs des contraintes du procédé, mais également les gradients de la fonction de coût et des contraintes. En présence d'erreurs structurales de modèle, ceci exige typiquement l'utilisation d'informations expérimentales pour le calcul des gradients du procédé. Des méthodes combinant estimation des paramètres et modification du problème d'optimisation, et employant de telles informations expérimentales, ont déjà été étudiées dans la littérature. Comme pour l'adaptation des contraintes, l'approche proposée dans cette thèse, nommée *adaptation des modificateurs*, présente l'avantage de ne pas nécessiter d'estimation paramétrique. En plus des modificateurs utilisés pour l'adaptation des contraintes, des termes de correction des gradients, basés sur la différence entre les gradients du procédé et ceux du modèle, sont ajoutés au coût et aux contraintes du problème d'optimisation. Il est montré que cette approche permet, sous réserve de convergence, d'atteindre un point qui satisfait les conditions nécessaires d'optimalité de premier ordre, pour le procédé réel. Cette approche est analysée en termes d'adéquation du modèle ; de plus, des conditions nécessaires de convergence et différentes stratégies de filtrage sont discutées. Les approches d'optimisation en temps réel par adaptation des contraintes et adaptation des modificateurs sont illustrées expérimentalement au moyen d'un système de trois réservoirs.

En pratique, les gradients ne sont pas mesurés et il est nécessaire de les estimer. Pour cela, des techniques de différences finies peuvent être employées. L'approche d'adaptation duale des modificateurs, étudiée dans cette thèse, conduit le procédé vers l'optimalité, tout en faisant attention à la précision de l'estimation des gradients, réalisée à partir de la séquence de points de fonctionnement. Une limite supérieure sur l'erreur d'estimation du gradient est déterminée théoriquement et est intégrée dans le problème d'optimisation sous la forme d'une contrainte, pour localiser le prochain point de fonctionnement.

Mots-clés : Optimisation en temps réel, adaptation des contraintes, régulation de contraintes, adaptation des modificateurs, estimation des gradients.

Contents

Frequently used Nomenclature	xvii
1 Introduction	1
1.1 Motivation	1
1.2 State of the Art	2
1.2.1 The Real-Time Optimization Paradigm	2
1.2.2 Real-Time Optimization as part of a Multilevel Control Structure	4
1.2.3 Input-Output Selection and Constraint Control	6
1.2.4 Real-Time Optimization with Model Update ...	8
1.2.5 Real-Time Optimization without Model Update	9
1.2.6 Determination of Plant Gradients	12
1.3 Scope, Organization and Contributions of the Thesis ..	13
2 Preliminaries	17
2.1 Static Optimization	17
2.1.1 Formulation of a NLP Problem	18
2.1.2 First-Order Necessary Conditions of Optimality	20
2.1.3 Second-Order Sufficient Conditions	21
2.1.4 Reduced Gradient and Hessian	22
2.1.5 Post-Optimal Sensitivity	23
2.2 Williams-Otto Reactor Problem	25
2.2.1 Description and Model Equations	25
2.2.2 Optimization Problem	28
2.3 Classical Two-Step Approach to RTO	29
2.3.1 Principles	29

2.3.2	Model-Adequacy Criterion	30
2.3.3	Numerical Illustration	33
2.4	ISOPE and Extensions	37
2.4.1	ISOPE	38
2.4.2	ISOPE with Model Shift	39
2.4.3	ISOPE with FFD Approach	39
2.4.4	Dual ISOPE	40
3	Real-Time Optimization via Constraint Adaptation	43
3.1	Variational Analysis of NCO in the Presence of Parametric Uncertainty	44
3.2	Constraint Adaptation	49
3.2.1	Principles of Constraint Adaptation	49
3.2.2	Constraint-Adaptation Algorithm	51
3.2.3	Feasibility	53
3.2.4	Active Set	53
3.2.5	Convergence	56
3.2.6	Effect of Measurement Noise	60
3.2.7	Alternative Constraint-Adaptation Algorithm	61
3.2.8	Case Study: Isothermal CSTR	64
3.3	Combination of Constraint Adaptation with Constraint Control	71
3.3.1	Constraint-Adaptation Scheme for Combination with MPC	72
3.3.2	Enforcing Constraints via MPC	76
3.3.3	Case Study: Planar Solid Oxide Fuel Cell System	81
3.4	Summary	95
4	Real-Time Optimization via Modifier Adaptation	99
4.1	Modifier Adaptation	100
4.1.1	Principles of Modifier Adaptation	100
4.1.2	Modifier-Adaptation Algorithm	103
4.1.3	KKT Matching	106
4.1.4	Convergence Analysis	109
4.1.5	Model Adequacy for Modifier-Adaptation Schemes	113
4.1.6	Alternative Schemes	114
4.1.7	Link to Previous Work	120
4.1.8	Case Study: Experimental Three-Tank System	120
4.2	Dual Modifier Adaptation	126

4.2.1	Estimation of Experimental Gradient from Past Operating Points	127
4.2.2	Dual Modifier Adaptation for Unconstrained Optimization	145
4.2.3	Dual Modifier Adaptation for Constrained Optimization	149
4.2.4	Case Study: Williams-Otto Reactor	152
4.3	Summary	157
5	Conclusions	161
5.1	Summary	161
5.2	Perspectives	163
A	Solid Oxide Fuel Cell Model	165
B	Affine Subspaces	171
	References	173

Frequently used Nomenclature

Abbreviations

CSTR	continuous stirred-tank reactor
DMC	dynamic matrix control
FFD	forward finite differencing
FU	fuel utilization
ISOPE	integrated system optimization and parameter estimation
KKT	Karush-Kuhn-Tucker
LICQ	linear-independence constraint qualification
LP	linear program
MPC	model predictive control
NCO	necessary conditions of optimality
NLP	nonlinear programming
NMPC	nonlinear model predictive control
QP	quadratic program
RTO	real-time optimization
SISO	single-input single-output
SOFC	solid oxide fuel cell

Operators

$\arg \min_x f(x)$	minimising argument of $f(x)$
$(\cdot)^{-1}$	inverse of a matrix
$(\cdot)^T$	transpose of a vector or matrix
$\text{diag}(a_1, \dots, a_n)$	diagonal matrix whose diagonal entries starting in the upper left corner are a_1, \dots, a_n
$\varrho\{\cdot\}$	spectral radius of a matrix
\inf	infimum
$\ \cdot\ $	norm of a vector or matrix

Subscripts, superscripts and indices

$(\cdot)^L$	lower bound
$(\cdot)^U$	upper bound
$(\cdot)^*$	optimal value
$(\cdot)_k$	value at RTO iteration k
$(\cdot)_m$	modified
$(\cdot)_p$	plant

Symbols

Latin symbols

Scalars

b_i	diagonal element of gain matrix \mathbf{B}
$g_i(\mathbf{u}, \mathbf{y})$	inequality constraint i
$G_{i,p}(\mathbf{u})$	inequality constraint i for the plant
$G_i(\mathbf{u}, \boldsymbol{\theta})$	inequality constraint i for the model
h	step size used in FFD
k	RTO iteration index
$L(\mathbf{u}, \boldsymbol{\theta})$	restricted Lagrangian function
n_g	dimension of vector $\mathbf{g}(\mathbf{u}, \mathbf{y})$
n_h	dimension of vector \mathbf{h}
n_K	dimension of vectors $\mathbf{\Lambda}$ and \mathbf{C} ($n_K = n_g + n_u(n_g + 1)$)
n_u	dimension of vector \mathbf{u}

n_x	dimension of vector \mathbf{x}
n_y	dimension of vector \mathbf{y}
n_z	dimension of vector $\mathbf{z}(\mathbf{u}, \boldsymbol{\theta})$ ($n_z = n_g + 2n_u$)
n_θ	dimension of vector $\boldsymbol{\theta}$
u_i	input or decision variable i
v	measurement noise
y_i	output variable i
$z_i(\mathbf{u}, \boldsymbol{\theta})$	inequality constraint or input bound i

Vectors

\mathbf{e}_j	j th unit vector
$\mathbf{g}(\mathbf{u}, \mathbf{y})$	inequality constraints
$\mathbf{G}_p(\mathbf{u})$	inequality constraints for the plant
$\mathbf{G}(\mathbf{u}, \boldsymbol{\theta})$	inequality constraints for the model
$\bar{\mathbf{g}}(\mathbf{u}, \mathbf{y})$	quantities being constrained ($\mathbf{g}(\mathbf{u}, \mathbf{y}) = \bar{\mathbf{g}}(\mathbf{u}, \mathbf{y}) - \bar{\mathbf{G}}^U$)
$\bar{\mathbf{G}}_p(\mathbf{u})$	plant quantities being constrained ($\mathbf{G}_p(\mathbf{u}) = \bar{\mathbf{G}}_p(\mathbf{u}) - \bar{\mathbf{G}}^U$)
$\bar{\mathbf{G}}(\mathbf{u}, \boldsymbol{\theta})$	model quantities being constrained ($\mathbf{G}(\mathbf{u}, \boldsymbol{\theta}) = \bar{\mathbf{G}}(\mathbf{u}, \boldsymbol{\theta}) - \bar{\mathbf{G}}^U$)
$\bar{\mathbf{h}}(\mathbf{u}, \mathbf{y})$	quantities being set via equalities
$\mathbf{H}_p(\mathbf{u})$	plant quantities being set via equalities
$\mathbf{H}(\mathbf{u}, \boldsymbol{\theta})$	model quantities being set via equalities
\mathbf{u}	inputs or decision variables
\mathbf{x}	state variables
\mathbf{y}	output variables
$\mathbf{z}(\mathbf{u}, \boldsymbol{\theta})$	inequality constraints including input bounds

Matrices

\mathbf{B}	gain matrix used to filter the constraint modifiers $\boldsymbol{\varepsilon}$
\mathbf{I}_n	identity matrix of dimension $n \times n$
\mathbf{K}	gain matrix

Sets

U	feasible set
U^*	optimal solution map

Greek symbols

Scalars

$\hat{\beta}_j$	j th entry of the gradient estimate $\hat{\beta}$
δ	measurement noise interval
v_i	Lagrange multiplier associated with the inequality constraint or input bound $z_i(\mathbf{u}, \boldsymbol{\theta})$
$\phi(\mathbf{u}, \mathbf{y})$	cost function
$\Phi_p(\mathbf{u})$	cost function for the plant
$\Phi(\mathbf{u}, \boldsymbol{\theta})$	cost function for the model
$\psi(\mathbf{u})$	general noisy function
$\Psi(\mathbf{u})$	general function
ε_i	constraint modifier i
$\hat{\varepsilon}_i$	auxiliary constraint modifier i
ε^{y_i}	modifier for model output i

Vectors and Matrices

$\hat{\beta}$	gradient estimate
$\boldsymbol{\gamma}^{\mathbf{G}}$	inequality constraint biases
$\boldsymbol{\gamma}^{\mathbf{H}}$	equality constraint biases
$\boldsymbol{\epsilon}(\mathbf{u})$	gradient error
$\boldsymbol{\epsilon}^n(\mathbf{u})$	gradient error due to measurement noise
$\boldsymbol{\epsilon}^t(\mathbf{u})$	gradient error due to truncation
$\boldsymbol{\zeta}^{\mathbf{U}}, \boldsymbol{\zeta}^{\mathbf{L}}$	Lagrange multipliers associated with the input bounds
$\boldsymbol{\theta}$	model parameters
$\boldsymbol{\lambda}$	ISOPE modifier
$\boldsymbol{\lambda}^{\mathbf{G}}$	constraint gradient modifiers
$\boldsymbol{\lambda}^{G_i}$	gradient modifier for constraint G_i
$\boldsymbol{\lambda}^{\Phi}$	cost gradient modifier
$\boldsymbol{\lambda}^{\mathbf{y}}$	gradient modifiers for the model outputs
$\boldsymbol{\lambda}^{y_i}$	gradient modifier for the model output i
$\boldsymbol{\Lambda}$	modifiers
$\hat{\boldsymbol{\Lambda}}$	auxiliary modifiers

$\boldsymbol{\mu}$	Lagrange multipliers associated with the inequality constraints $\mathbf{G}(\mathbf{u}, \boldsymbol{\theta})$
\boldsymbol{v}	Lagrange multipliers associated with the collected inequality constraints and input bounds $\mathbf{z}(\mathbf{u}, \boldsymbol{\theta})$
ε	constraint modifiers
$\hat{\varepsilon}$	auxiliary constraint modifiers
ε^y	model output modifiers

Calligraphic symbols

Scalars

$\mathcal{L}(\mathbf{u}, \boldsymbol{\mu}, \boldsymbol{\zeta}^U, \boldsymbol{\zeta}^L, \boldsymbol{\theta})$ Lagrangian function

Vectors

$\mathcal{C}_p(\mathbf{u})$ KKT-related quantities for the plant
 $\mathcal{C}(\mathbf{u}, \boldsymbol{\theta})$ KKT-related quantities for the model
 $\mathcal{M}(\varepsilon_k)$ constraint-adaptation algorithm
 $\mathcal{M}(\varepsilon_k)$ modifier-adaptation algorithm

Sets

\mathcal{A} active set

Introduction

1.1 Motivation

It is sufficient to type the words “real-time optimization” on an Internet search engine and to navigate through the generated results to realize the tremendous importance of real-time optimization (RTO) in today’s scenario, both in industry and in the research community. Nowadays, RTO implementations are proposed for a wide variety of applications that go far beyond the initial applications in the chemical and petrochemical industries. These include the RTO of food production processes, biological processes, pulp and paper production, mineral production, etc. Also, numerous companies offer RTO solutions and related software. The main reason for such success is that RTO can provide significant economic payoffs. For a high capacity plant, even a 1% improvement in yield can lead to significant annual savings [21]. RTO is targeted at reducing costs and improving profitability while taking into account security, quality, environmental and equipment constraints. These are the very issues that are of capital importance in today’s highly competitive markets, even in times of economic slowdown.

RTO relies on a process model that is used to compute optimal operating conditions. One of the main challenges in RTO systems comes from the fact that models are simplified representations of the reality and are thus subject to uncertainty. With the availability of plant operating data, the process model can be updated in order to give a better prediction of the plant outputs. In the “classical” iterative two-step approach, a parameter estimation step is performed in order to update

the model, which is subsequently used in the optimization step to compute the new operating point. As one might expect, the performance of RTO systems depends on how accurately the model represents the process behavior [35, 91]. However, accurate models are difficult to obtain with affordable effort. In the presence of plant-model mismatch and unmeasured disturbances, the solution provided by the real-time optimizer might result in suboptimal operation for the plant. Worse, if the constraints are not well predicted by the model, an infeasible solution might result. Furthermore, it is not straightforward to decide which model parameters to adapt by means of parameter estimation, and which ones to keep at fixed values [97]. The key model parameters should be identifiable from the available measurements, which is not always the case [89]. Parameter estimation is further complicated by the nonlinearity of the model, poor quality of the data, and the lack of tight bounds on the parameter search space [49]. Hence the interest in devising and studying RTO methodologies that are tailored to enforcing feasibility and optimality, while alleviating model accuracy and model updating requirements. With this in mind, this thesis extends and formalizes several ad hoc techniques that are available in the area of real-time optimization.

1.2 State of the Art

1.2.1 The Real-Time Optimization Paradigm

The chemical industry is characterized by continuous operating plants operating at near steady-state. The synthesis of control structures for these plants presents several difficulties which can be summarized as follows:

- Imperfect models
- Nonlinearities
- Presence of operating constraints
 - *Equipment constraints*: Pressure drop through equipment, valve positions, compressor speeds, horsepower limits, column loading, etc.
 - *Safety constraints*: Explosive limits, critical pressure and temperatures.
 - *Quality constraints*: Reaction yields and selectivities, product purity, etc.

- *Environmental constraints*: CO₂ emissions, pollution levels, etc.
- Multivariable nature
- Measurement noise and process disturbances
- Changing optimum due to:
 - *Process degradation*: From the start-up of a process until its shutdown for maintenance, the process goes through a continuous state of degradation, including wearing of mechanical equipment, fouling of heat exchangers, decaying of catalyst activities, plugging of lines and distributors, fouling of turbines, bypassing in packed beds and reactors due to fouling, etc.
 - *Weather conditions*: External weather conditions affect cooling water temperatures, air cooler efficiencies and heat loss from equipment.
 - *Changes in production goals*: Production objectives change with the market demands.

In response to these difficulties, the use of process control and optimization technology has become the rule for a large number of chemical and petrochemical plants. The advantages of on-line computer-aided process operation are largely recognized [21].

In any RTO application, the optimum plant operating conditions may change as a result of process degradation, weather conditions, and changes in production goals. The RTO system attempts to track the plant optimum changing at low frequency to maintain the plant at its most profitable operating point [98]. Real-time optimization, on-line optimization and optimizing control are different terms that have been used in the literature to designate the same purpose, which is the continuous reevaluation and alteration of operating conditions of a process so that economic productivity is maximized subject to operational constraints [2, 46, 63].

RTO emerged in the chemical process industry in the seventies together with model predictive control (MPC), at the time when on-line computer control of chemical plants became available. Since then, there has been extensive research in the area of RTO, and numerous industrial applications have been reported. A number of successful industrial applications is listed in [63]. Today RTO is a very extensive research field, strongly connected with other fields such as input-output selection, MPC, parameter estimation, state estimation, data reconciliation and results analysis.

1.2.2 Real-Time Optimization as part of a Multilevel Control Structure

In highly automated plants, the goal of an economically optimal operation is typically addressed by an automation decision hierarchy involving several levels, as shown in Figure 1.1 [23, 63, 92]. This is essentially a cascade structure, which is in general appropriate because of the difference in time scale associated with the disturbance frequencies and decisions at each level [63].

At the upper level, the planning and scheduling addresses long term issues like production rate targets and raw material allocation. The disturbances associated with this level are market fluctuations, demand and price variations. At the lowest level, sensor measurements are collected from the plant, and basic flow, pressure, and temperature control is implemented, possibly via advanced regulatory controllers that include cascade controllers and multivariable controllers. Linear model predictive controllers (MPC) have gained wide acceptance because they can handle large multivariable systems with operating constraints [70]. The process control layer achieves safety, product quality, production rate goals and stable operation. The associated time scale is usually in the order of seconds to minutes.

The RTO level provides the bridge between plant scheduling and process control. At this level, medium-term decisions are made on a time scale of hours to days by considering economics explicitly in operations decisions. This step typically relies on a real-time optimizer, which determines the optimal operating point under changing conditions, e.g., low frequency process disturbances such as catalyst decay or changes in raw material quality. The operating point is characterized by setpoints for a set of controlled variables that are passed to the lower-level controllers. Model-based RTO typically involves nonlinear first-principle models describing the steady-state behavior of the plant. These models are often relatively large, and the model-based optimization may require substantial computing time [63].

Uncertainty is present in the form of plant-model mismatch and unmeasured disturbances. In order to combat uncertainty, all levels utilize process measurements as inputs to their feedback loops, and each higher level provides guidance to the subsequent lower level.

The multilevel structure leads to a vertical decomposition of the automation tasks. There is a second type of decomposition that is done horizontally, and is known as multiechelon decomposition [65].

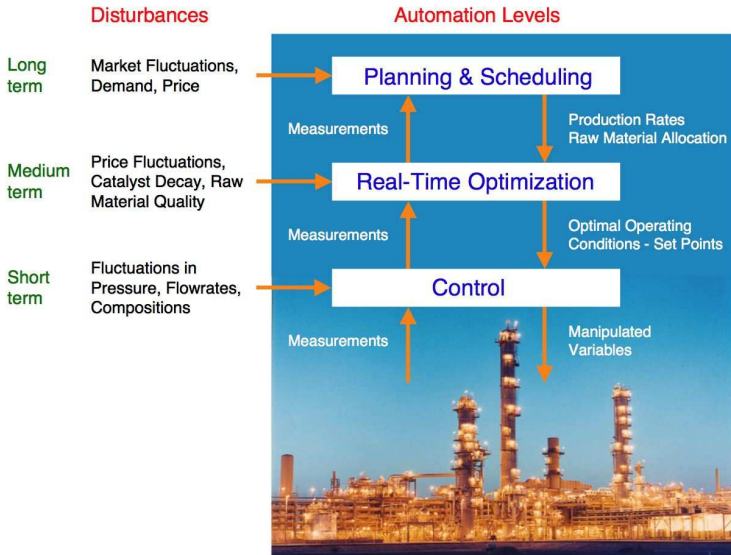


Fig. 1.1. Automation Decision Hierarchy.

This decomposition gives rise to the approaches known as decentralized optimizing control [65], optimizing control for integrated plants [3], optimizing control of interconnected systems [13], or plant-wide control [52, 55, 80]. The plant is divided into interacting groups of processing units, for which the optimization is carried out separately. These are coordinated from time to time by a coordinator.

The multilevel structure given in Figure 1.1 has some drawbacks. The main disadvantage is that sampling and optimization have to be delayed until the controlled plant has settled to a new steady state [29]. This delay occurs at each RTO step after a change in the input variables, and worse, after the occurrence of disturbances. As the optimization is only performed intermittently at a rate such that the controlled plant is assumed to be at (pseudo) steady state, the adaptation of the operating conditions can be slow. Also, inconsistencies may arise from the use of different models at the different levels. Schemes aiming to reduce the gap between the regulation and RTO levels have been proposed in order to address these issues. A review of these schemes is given in [29]. Notice that, for the case of measured disturbances and known changes in operating conditions, it is possible to carry out

the optimization using the current steady-state model without waiting for steady state [77]. The model parameters are only updated using measurements obtained at steady-state. However, the optimization is carried out at a much faster rate using the updated disturbance values.

For applications exhibiting strongly nonlinear and complex dynamics, the replacement of linear MPC controllers by nonlinear model-predictive controllers (NMPC), for which the RTO and regulatory levels are completely merged into a single level, is gaining interest in the research community [4, 50, 67, 84]. One of the advantages of NMPC is that it can react quickly to disturbances since it does not require to wait for the controlled plant to reach a steady state. Also, no inconsistencies arise from the use of different models on different layers [29]. The use of a rigorous first-principle prediction model has the disadvantage, however, of requiring the solution of an optimal control problem in real-time [25]. Computational complexities, the requirement of real-time identification techniques for nonlinear processes, and the lack of stability and robustness results for the case of nonlinear systems, are important limitations to a practical implementation of NMPC [16, 50].

1.2.3 Input-Output Selection and Constraint Control

The selection of manipulated and controlled variables for the regulatory level, as well as the choice of decision (input) variables at the optimization level, is not unique. Several authors have treated the relative importance of control objectives in the synthesis of optimizing control structures for chemical processes (see e.g. [65, 80] and references therein). In this context, operating constraints have a fundamental role to play.

In RTO systems, high-priority objectives such as safety, product quality and production rates are often formulated as equality constraints. It is generally recognized that equality constraints should always be controlled. A number of manipulated variables are assigned to keep these constraints tight. The degrees of freedom available for optimization are the remaining ones that are not used to control the equality constraints [63].

Inequality constraints can be active or inactive at the optimum. If some constraints are known to be active irrespective of the process operating conditions and the disturbances entering the process, these constraints can be controlled all the time and treated as equality constraints with selected manipulated variables being assigned to keeping

them active [56]. Heuristics have been proposed for identifying which constraints are active for different types of processes [32].

Other constraints might become critical depending on the process conditions and disturbances. This means that the optimal operating point can switch from the intersection of one set of active constraints to another, as process disturbances change with time. When this occurs, the control system should be capable of automatically switching to the new active set. This is referred to as constraint control [56]. Constraint control requires the controlled constrained quantities to be measured (or estimated) online. The advantages of controlling the constraints are [2, 40]:

1. Constraint control avoids infeasibilities due to short-term disturbances and during the dynamic regimes that take place when the plant is not operating at a steady state.
2. If the active constraints are regulated, the need for on-line optimization may be eliminated for a certain set of middle-term disturbances, for which the set of constraints that determine the optimum does not change.
3. The control of active constraints can be implemented with minimal modeling using direct process measurement. Offset-free behavior can be achieved despite the absence of a process model.

Changes in the active set due to different operating conditions have been considered in [56]. When a constraint becomes active, the feedback controllers override each other. Single-input single-output (SISO) loops are employed, although it is suggested that, in case of severe interaction, some form of non-interacting control be applied. The loops for each possible active set are predetermined.

An approach on how to handle changing active sets in the regulatory level by altering the regulatory control structure is presented in [2]. In order to move the operation towards the new optimum, which is determined by a new set of active constraints, the constraint control loops are partitioned into servo loops and regulation loops. The constraint setpoints of the servo loops are changed, while keeping the regulated constraints tight. Using this approach, alternative operational routes to the new optimum are possible, each route involving different sequencing of control structures. Screening criteria are given to select the best feasible sequencing route.

1.2.4 Real-Time Optimization with Model Update

Since accurate models are rarely available in industrial applications, RTO typically proceeds by an iterative two-step approach [20, 46, 63], 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 (adjustable) model parameters for which the (steady-state) model gives a good prediction of the measured plant outputs at the current operating point. In the optimization step, the updated model is used to determine a new operating point by solving a model-based optimization problem that typically consists in a nonlinear programming (NLP) problem. Besides model update and optimization, RTO systems also encompass subsystems for data validation and results analysis. A typical closed-loop RTO system is represented in Figure 1.2. The performance of the overall RTO system will depend on the design of the sub-systems [34, 63, 74]. Important design decisions include the selection of the decision variables at the RTO level and of the controlled variables at the process control level, measurement selection, partitioning of the model parameters into those that will be adjusted online, and those that will be fixed at nominal values, model structure, model updating approach and results analysis methods.

Since the models used in most RTO applications are nonlinear steady-state models, it is important to verify that the plant is near steady-state operation before carrying out data validation and updating the model. The validation of process measurements includes steady-state detection, gross error detection, and data reconciliation. Steady-state detection is complicated by the fact that different sections of the plant may be at different transient states. Data reconciliation uses material and energy balances to ensure that the data used for model update is self-consistent.

The classical two-step approach works well provided that (i) there is little structural plant-model mismatch [91], 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 necessary conditions of optimality (NCO) of the plant simply by estimating the model parameters that predict the plant outputs well. Some information regarding plant gradients needs to be incorporated into the RTO

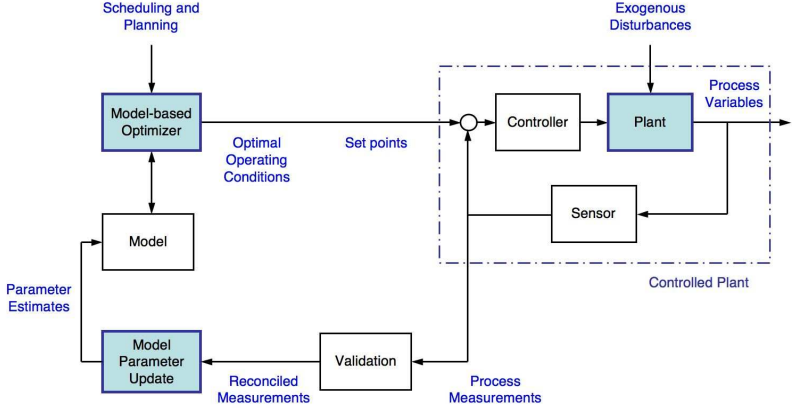


Fig. 1.2. Closed-loop RTO system with model update [98].

scheme. In the so-called *integrated system optimization and parameter estimation (ISOPE) method* [13, 74], a gradient-modification term is added to the cost function of the optimization problem to force the iterations to converge to a point that satisfies the plant NCO. Regarding (ii), the use of multiple data sets has been suggested to increase the number of identifiable parameters [89]. How to select the additional data sets based on the design of plant experiments has also been addressed [90].

In response to both (i) and (ii), methods that do not rely on a model update based on a parameter estimation problem have gained in popularity in recent years.

1.2.5 Real-Time Optimization without Model Update

This class of methods encompasses the model-free and fixed-model methods that are discussed next.

Model-Free Methods

Model-free methods do not use a process model on-line for implementing the optimization. Two classes of approaches can be further distinguished. In the first one, successive operating points are determined by “mimicking” iterative numerical optimization algorithms.

For example, evolutionary-like schemes have been proposed that implement the Nelder-Mead simplex algorithm to approach the optimum [9]. To achieve faster convergence rates, techniques based on gradient-based algorithms have also been developed [40]. The second approach to model-free methods consists in recasting the NLP problem into that of choosing outputs whose optimal values are approximately invariant to uncertainty. The idea is then to use measurements to bring these outputs to their invariant values, thereby rejecting the uncertainty. In other words, a feedback law is sought that implicitly solves the optimization problem, as it is done in self-optimizing control [78], and NCO tracking [36].

Fixed-Model Methods

Fixed-model methods utilize both a nominal process model and appropriate measurements for guiding the iterative scheme towards the optimum. The process model is embedded within an NLP problem that is solved repeatedly. But instead of refining the parameters of a first-principles model from one RTO iteration to the next, the measurements are used to update the cost and constraint functions in the optimization problem so as to yield a better approximation of the plant cost and constraints at the current point. Fixed-model RTO methods encompass the *bias-update approach* [33], and the recent approach by *Gao and Engell* [37].

Bias-update Approach. Optimizing process performance in the presence of constraints requires the use of a mathematical model of the process to anticipate the effect of the constraints and take appropriate control actions [15]. Thus, model-based control systems such as dynamic matrix control (DMC) [22], model algorithmic control (MAC) [71], and internal model control (IMC) [39] provide a natural setting for dealing with constraints. All the forgoing make use of plant-model biases. The manipulated variables are driven by the model, while the effects of disturbances are taken into account by subtracting the model outputs from the measured plant outputs. These methods differ only in the design of the controller. For discrete-time control of linear systems, it has been shown that plant-model biases yield interesting properties such as zero offset and robustness to modeling errors [41]. The success of the bias-update method at the control level motivates its incorporation into the steady-state optimization level. However, the analysis

of bias update at the steady-state optimization level has received little attention in the research community. In [33], adjustable bias terms are added to the equality and inequality constraints of the steady-state RTO problem. These constraint-correction factors, or modifiers, are obtained as the difference between the measured and predicted values of the constraints. The analysis undertaken in [33] is restricted to problems where there are as many independent, active constraints as decision variables at the plant optimum. For this particular case, the analysis in [33] considers how model adequacy in the sense of Forbes et. al [35] (see Subsection 2.3.2) applies to the bias-update case. Model adequacy is shown to be determined by the gradient of the cost function and of the active constraints predicted by the model. It is also shown that, when the constraints are affine and the model is adequate, the bias-update scheme will converge to the true process optimum (which by assumption is completely determined by the active constraints) provided the filter is designed to ensure stability. A similar scheme has also been presented in [24] where the approach, referred to as *IMC optimization*, is applied using two models. The computational burden of performing a direct optimization based on a complex non-linear model is reduced by performing the optimization on a reduced (internal) model that is embedded in an IMC optimization structure. The solution, however, may be sub-optimal. Recently, a bias-update approach was applied in the RTO of a pulp mill benchmark problem using a reduced linear model [64].

Bias and Gradient-update Approach. A modified ISOPE approach, which eliminates the need to estimate the model parameters, was proposed by Tatjewski [82]. Following this idea, Gao and Engell [37] recently formulated a modified optimization problem, wherein both the constrained values and the cost and constraint gradients are corrected. The main contribution in [37] lies in the way in which the process constraints are corrected. In addition to the correction term for the constraint values used in bias update [17, 33], gradient-correction terms are added to the constraints. These are based on the difference between the estimated and predicted values of the constraint gradients. This way, an operating point that satisfies the plant NCO is obtained upon convergence.

Comparisons between different RTO approaches have been published in [28], and more recently in [18, 98].

1.2.6 Determination of Plant Gradients

The major difficulty in the implementation of ISOPE approaches, and of the modifier-adaptation approach studied in this thesis, is the determination of the gradient of the plant outputs with respect to the inputs, also called experimental gradients. In the context of RTO, several methods for estimating experimental gradients have been proposed. These methods have been compared by several authors [58, 98]. A distinction can be made between *steady-state perturbation methods* and *dynamic perturbation methods*. Steady-state perturbation methods require at least $(n_u + 1)$ steady-state operating points in order to estimate the gradients, where n_u is the number of decision variables. The idea behind dynamic perturbation methods is to estimate the experimental gradient using information obtained during the transition between steady-state operating points corresponding to successive RTO iterations. One such approach is dynamic model identification, where the experimental gradients are obtained from a dynamic model that is identified during transient operation [5, 38, 93, 98]. The major advantage of dynamic perturbation methods is that the waiting time needed to reach a new steady state can be avoided. However, they have the disadvantage of requiring additional excitation to identify the dynamic model.

Regarding steady-state perturbation methods, the most straightforward approach to estimate experimental gradients is to apply finite-difference techniques directly on the plant. These techniques consist in perturbing each input variable individually around the current operating point to get an estimate of the corresponding gradient from the measured plant outputs at steady state. The use of finite differences was suggested in the original ISOPE paper [72]. However, since a new steady state must be attained for each perturbation in order to evaluate the plant derivatives, the time between successive RTO iterations increases significantly, and the approach becomes experimentally intractable with several input variables.

Other steady-state perturbation methods use the current and past operating points to estimate the gradients. In the so-called dual ISOPE algorithms [13, 81], a constraint on the search region for the next operating point, which takes into account the position of the past ones, is added to the model-based optimization problem. This way, the optimization objective and the requirement of generating sufficient steady-

state perturbations in the operating points for the purpose of estimating the gradient are dealt with simultaneously.

1.3 Scope, Organization and Contributions of the Thesis

This thesis is dedicated to the study of the fixed-model RTO methods described in Subsection 1.2.5. It considers only the optimizing control of a single unit (or of the whole plant considered as an integrated system). The characteristics of decentralized optimizing control are not discussed. The reader interested in this topic is directed to the literature referenced in Subsection 1.2.2. Data validation and results analysis are not discussed either, a description of these RTO components and reference to related bibliography is given in [63]. An RTO application where data validation with steady state detection and gross error detection are discussed in detail can be found in [99].

After a brief description of the state of the art in RTO in connexion with the approaches studied in this thesis, a number of preliminary results are given in Chapter 2.

Chapter 2: Preliminaries. The RTO static optimization problem is formulated as a nonlinear program (NLP). Theoretical results on NLP are presented, which will be used in the theoretical analysis carried out in subsequent chapters. The effect of plant-model mismatch in the RTO closed-loop system of Figure 1.2 is discussed, model-adequacy is reviewed, and the ISOPE approach tailored to deal with plant-model mismatch is revisited.

The main contributions of this thesis to the RTO literature can be found in Chapters 3 and 4, which are summarized next:

Chapter 3: Real-Time Optimization via Constraint Adaptation. In the context of this thesis, the correction of the constraint functions in the model-based optimization problem, using bias updates as in [33], is referred to as *constraint adaptation* [17, 60]. As discussed in Subsection 1.2.5, the analysis of the merits of constraint adaptation has received little attention in the RTO literature. This chapter undertakes a novel study of various aspects of constraint adaptation,

including the necessary conditions for convergence. Unlike the analysis undertaken in [33], the analysis in Chapter 3 is not restricted to problems where the optimal solution is completely determined by the active constraints. Furthermore, the analysis in Chapter 3 is made independently of which constraints become active. Several illustrative examples are studied. A strategy for integrating constraint adaptation at the RTO level with MPC at the process control level is presented. The MPC is tailored to control the constraints. The integration strategy includes two important features: (i) a novel way of adapting the constraints permits to update the constraint setpoints passed to the constraint controller at each iteration. The actual constraint bounds for the active constraints are reached upon convergence. And (ii), the residual degrees of freedom are exploited by enforcing the inputs to their optimal values along selected directions in the input space. In the context of a collaboration with the *Laboratoire d'Énergétique Industrielle of EPFL (LENI)*, the constraint-adaptation approach is applied in simulation to a solid-oxide fuel cell stack [62]. An experimental validation of the results is foreseen for an experimental SOFC system available at LENI.

Chapter 4: Real-Time Optimization via Modifier Adaptation.

In this chapter, additional modifiers are introduced into the optimization problem in order to correct the gradients of the cost and constraint functions. These gradient corrections require information on the plant gradients that should be available at every RTO period. The approach used to modify the cost and constraint functions is similar to the approach recently proposed in [37] for the modification of the constraint functions. This chapter formalizes the idea of using plant measurements to adapt the optimization problem in response to plant-model mismatch, following the paradigm of modifier adaptation. Model-adequacy requirements and local convergence conditions are discussed. Two filtering strategies are proposed to achieve stability and reduce sensitivity to measurement noise. Links between modifier adaptation and related work in the literature are highlighted. The applicability of the constraint-adaptation and modifier-adaptation approaches is shown through an experimental application to a three-tank system. Following the ideas advanced by the dual ISOPE approach, a dual modifier-adaptation approach is proposed in order to estimate experimental gradients reliably from past operating points while updating the operating point. The error in the gradient estimates ob-

tained from past operating points is analyzed. Based on this analysis, a norm-based constraint is formulated. This constraint is incorporated into the dual modifier-adaptation scheme.

Finally, Chapter 5 concludes the thesis.

Chapter 5: Conclusions. This chapter concludes the thesis and discusses the perspectives in terms of new research topics.

The constraint adaptation approach has also been extended to batch processes in [59]. However, this extension falls outside the scope of this thesis.

Concerning the strategy for integrating constraint adaptation with an MPC constraint controller proposed in Chapter 3, the emphasis is not on MPC, but on the integration strategy from a methodological point of view. The analysis of the integrated scheme, as well as a comparison with other strategies such as LP-MPC and QP-MPC [88], are beyond the scope of this thesis.

In order to estimate the experimental gradients required by modifier adaptation, only steady-state perturbation approaches are considered in this thesis. No attempt is made to compare steady-state perturbation methods with dynamic perturbation methods, which fall outside the scope of this thesis.

Preliminaries

The optimization of steady-state operating conditions involves static optimization as opposed to dynamic optimization. This chapter begins by presenting several preliminary results on static optimization in Section 2.1. The static optimization problem is formulated as a nonlinear programming (NLP) problem, for which the first-order necessary conditions of optimality (NCO) and the second-order sufficient conditions are presented, as well as first-order sensitivity results based on second-order assumptions. A case study corresponding to a continuous reactor example is presented in Subsection 2.2, which includes a simple, but realistic, example of structural modeling error. This example is used in Section 2.3 to illustrate some of the difficulties of the classical two-step approach to RTO (see Subsection 1.2.4). Model-adequacy conditions proposed in the literature are also discussed in Section 2.3.

Next, the integrated system optimization and parameter estimation (ISOPE) approach, is presented in Section 2.4, as well as a few of its variants. Their description will help understand the link and the differences between ISOPE and the modifier-adaptation approach presented in Chapter 4. The dual ISOPE approach presented in Subsection 2.4.4 is the motivator for the dual modifier-adaptation approach of Chapter 4.

2.1 Static Optimization

The usual objective in RTO is the minimization or maximization of some steady-state operating performance of the process (e.g., mini-

mization of operating cost or maximization of production rate), while satisfying a number of constraints (e.g., bounds on process variables or product specifications).

Throughout this thesis, for simplicity, only inequality constraints are considered in the RTO problem. The inclusion of equality constraints to the constraint-adaptation and modifier-adaptation schemes studied in this thesis poses no conceptual difficulty, since equality constraints would behave like inequality constraints that are always active. However, equality constraints are explicitly considered in Subsection 3.3.1, where constraint adaptation is combined with constraint control.

2.1.1 Formulation of a NLP Problem

In the context of RTO, since it is important to distinguish between the plant and the model, we will use the notation $(\cdot)_p$ for the variables associated with the plant.

The steady-state optimization problem for the plant is formulated as follows:

$$\begin{aligned} \mathbf{u}_p^* \in \arg \min_{\mathbf{u}} \quad & \phi(\mathbf{u}, \mathbf{y}_p) \\ \text{s.t.} \quad & \mathbf{g}(\mathbf{u}, \mathbf{y}_p) \leq \mathbf{0} \\ & \mathbf{u}^L \leq \mathbf{u} \leq \mathbf{u}^U, \end{aligned} \tag{2.1}$$

where $\mathbf{u} \in \mathbb{R}^{n_u}$ denote the decision (or input) variables, and $\mathbf{y}_p \in \mathbb{R}^{n_y}$ are the measured (or output) variables; $\phi : \mathbb{R}^{n_u} \times \mathbb{R}^{n_y} \rightarrow \mathbb{R}$ is the scalar cost function to be minimized; $g_i : \mathbb{R}^{n_u} \times \mathbb{R}^{n_y} \rightarrow \mathbb{R}$, $i = 1, \dots, n_g$, is the set of inequality constraint functions; and \mathbf{u}^L , \mathbf{u}^U are the bounds on the decisions variables.

An equivalent formulation of the optimization problem (2.1) is as follows:

$$\begin{aligned} \mathbf{u}_p^* \in \arg \min_{\mathbf{u}} \quad & \Phi_p(\mathbf{u}) \\ \text{s.t.} \quad & \mathbf{G}_p(\mathbf{u}) \leq \mathbf{0} \\ & \mathbf{u}^L \leq \mathbf{u} \leq \mathbf{u}^U, \end{aligned} \tag{2.2}$$

with Φ_p and \mathbf{G}_p being defined as $\Phi_p(\mathbf{u}) := \phi(\mathbf{u}, \mathbf{y}_p(\mathbf{u}))$ and $\mathbf{G}_p(\mathbf{u}) := \mathbf{g}(\mathbf{u}, \mathbf{y}_p(\mathbf{u}))$.

In any practical application, the input-output mapping representing the plant operation at steady-state, $\mathbf{y}_p(\mathbf{u})$, is not known precisely, as only an approximate model is available:

$$\begin{aligned}\mathbf{f}(\mathbf{u}, \mathbf{x}, \boldsymbol{\theta}) &= \mathbf{0} \\ \mathbf{y} &= \mathcal{F}(\mathbf{u}, \mathbf{x}, \boldsymbol{\theta}),\end{aligned}$$

where $\mathbf{f} \in \mathbf{R}^{n_x}$ is the set of process model equations including mass and energy balances, thermodynamic relationships, etc., $\mathbf{x} \in \mathbf{R}^{n_x}$ are the model states, $\mathbf{y} \in \mathbf{R}^{n_y}$ are the output variables predicted by the model, which are evaluated through the functions \mathcal{F} , and $\boldsymbol{\theta} \in \mathbf{R}^{n_\theta}$ is the set of adjustable model parameters. For simplicity, we shall assume that the model outputs \mathbf{y} can be expressed explicitly as functions of \mathbf{u} and $\boldsymbol{\theta}$ only, $\mathbf{y}(\mathbf{u}, \boldsymbol{\theta})$. Thereby, the solution to the original problem (2.2) can be approached by solving the following NLP problem:

$$\begin{aligned}\mathbf{u}^* \in \underset{\mathbf{u}}{\operatorname{arg\,min}} \quad & \Phi(\mathbf{u}, \boldsymbol{\theta}) & (2.3) \\ \text{s.t.} \quad & \mathbf{G}(\mathbf{u}, \boldsymbol{\theta}) \leq \mathbf{0} \\ & \mathbf{u}^L \leq \mathbf{u} \leq \mathbf{u}^U,\end{aligned}$$

with Φ and \mathbf{G} defined as $\Phi(\mathbf{u}, \boldsymbol{\theta}) := \phi(\mathbf{u}, \mathbf{y}(\mathbf{u}, \boldsymbol{\theta}))$ and $\mathbf{G}(\mathbf{u}, \boldsymbol{\theta}) := \mathbf{g}(\mathbf{u}, \mathbf{y}(\mathbf{u}, \boldsymbol{\theta}))$.

This formulation assumes that $\phi(\mathbf{u}, \mathbf{y}_p)$ and $\mathbf{g}(\mathbf{u}, \mathbf{y}_p)$ are known functions of \mathbf{u} and \mathbf{y}_p , i.e., they can be evaluated directly from the measurements. On the other hand, the dependency on the model parameters $\boldsymbol{\theta}$ of the cost and constraint values predicted by the model, $\phi(\mathbf{u}, \mathbf{y})$ and $\mathbf{g}(\mathbf{u}, \mathbf{y})$, is implicit via the model outputs $\mathbf{y}(\mathbf{u}, \boldsymbol{\theta})$.

It is assumed throughout that Φ and \mathbf{G} are twice continuously differentiable with respect to \mathbf{u} . Under the assumptions that the feasible set $U := \{\mathbf{u} \in [\mathbf{u}^L, \mathbf{u}^U] : \mathbf{G}(\mathbf{u}, \boldsymbol{\theta}) \leq \mathbf{0}\}$ is nonempty for $\boldsymbol{\theta}$ given, the infimum for Problem (2.3), labeled Φ^* , is assumed somewhere on $[\mathbf{u}^L, \mathbf{u}^U]$ and thus a minimum exists by Weierstrass' theorem (see, e.g., [7], Theorem 2.3.1). Here, we shall denote by \mathbf{u}^* a minimizing solution for Problem (2.3), i.e., $\Phi(\mathbf{u}^*, \boldsymbol{\theta}) = \Phi^*$, and by $\mathcal{A} := \{i : G_i(\mathbf{u}^*, \boldsymbol{\theta}) = 0, i = 1, \dots, n_g\}$ the set of active inequality constraints at \mathbf{u}^* . In the presence of plant-model mismatch, a model-based solution \mathbf{u}^* does not in general match the plant optimum \mathbf{u}_p^* .

Although the emphasis in this thesis is placed on the RTO of continuous processes operating at steady state, it is worth recalling that the static optimization problem formulated above is also encountered in the framework of run-to-run optimization of batch and semi-batch processes [36]. The optimization of batch and semi-batch processes typically involves solving a dynamic optimization problem with path and terminal constraints. The solution consists of time-varying input profiles, $\mathbf{m}(t)$. Numerically, such problems can be solved by parameterizing the input profiles using a finite number of parameters \mathbf{u} , e.g., piecewise polynomial approximations of $\mathbf{m}(t)$. This way, the inputs profiles can be expressed as $\mathbf{m}(t) = \mathcal{U}(\mathbf{u}, t)$, and the optimization is performed with respect to \mathbf{u} . Although the process is dynamic in nature, a static map can then be used to describe the relationship between the inputs \mathbf{u} and the outcome of the batch \mathbf{y}_p . Making use of this static map, the dynamic optimization problem can be reformulated into a static optimization problem similar to (2.2) [36].

In the context of this thesis, a distinction is made between model inaccuracies due to structural errors and parametric uncertainties. A process model is said to be structurally correct if there exist values of the adjustable model parameters $\boldsymbol{\theta}$ such that the model yields a precise representation of the input-output mapping of the plant. This is formalized in the definition below:

Definition 2.1 (Structural Model Errors)

A process model $\mathbf{y}(\mathbf{u}, \boldsymbol{\theta})$ is said to be structurally correct, if there exist values of the adjustable model parameters $\boldsymbol{\theta}$ such that $\mathbf{y}(\mathbf{u}, \boldsymbol{\theta}) = \mathbf{y}_p(\mathbf{u})$, $\frac{\partial \mathbf{y}}{\partial \mathbf{u}}(\mathbf{u}, \boldsymbol{\theta}) = \frac{\partial \mathbf{y}_p}{\partial \mathbf{u}}(\mathbf{u})$, and $\frac{\partial^2 \mathbf{y}}{\partial \mathbf{u}^2}(\mathbf{u}, \boldsymbol{\theta}) = \frac{\partial^2 \mathbf{y}_p}{\partial \mathbf{u}^2}(\mathbf{u})$, for all $\mathbf{u} \in U$. On the other hand, if such parameter values do not exist, the process model is said to be structurally incorrect.

Notice that, for all points \mathbf{u} in the interior of U (not belonging to the boundary of U), the condition $\mathbf{y}(\mathbf{u}, \boldsymbol{\theta}) = \mathbf{y}_p(\mathbf{u})$ for all $\mathbf{u} \in U$ implies $\frac{\partial \mathbf{y}}{\partial \mathbf{u}}(\mathbf{u}, \boldsymbol{\theta}) = \frac{\partial \mathbf{y}_p}{\partial \mathbf{u}}(\mathbf{u})$, and $\frac{\partial^2 \mathbf{y}}{\partial \mathbf{u}^2}(\mathbf{u}, \boldsymbol{\theta}) = \frac{\partial^2 \mathbf{y}_p}{\partial \mathbf{u}^2}(\mathbf{u})$.

2.1.2 First-Order Necessary Conditions of Optimality

Theorem 2.1 (Karush-Kuhn-Tucker Necessary Conditions)

Let \mathbf{u}^ be a (local) optimum of Problem (2.3) and assume that the linear-independence constraint qualification (LICQ) holds at \mathbf{u}^* :*

$$\text{rank} \left(\begin{array}{ccc} \frac{\partial \mathbf{G}}{\partial \mathbf{u}} & \text{diag}(\mathbf{G}) & \\ \mathbf{I}_{n_u} & & \text{diag}(\mathbf{u} - \mathbf{u}^U) \\ -\mathbf{I}_{n_u} & & \text{diag}(\mathbf{u}^L - \mathbf{u}) \end{array} \right)_{\mathbf{u}^*, \boldsymbol{\theta}} = n_g + 2n_u \quad (2.4)$$

i.e., this $(n_g + 2n_u) \times (n_g + 3n_u)$ matrix is of full row rank. Then, there exist (unique) values for the Lagrange multiplier vectors $\boldsymbol{\mu} \in \mathbf{R}^{n_g}$, $\boldsymbol{\zeta}^U, \boldsymbol{\zeta}^L \in \mathbf{R}^{n_u}$ such that the following first-order Karush-Kuhn-Tucker (KKT) conditions hold at \mathbf{u}^* :

$$\mathbf{G} \leq \mathbf{0}, \quad \mathbf{u}^L \leq \mathbf{u} \leq \mathbf{u}^U \quad (2.5)$$

$$\boldsymbol{\mu}^\top \mathbf{G} = 0, \quad \boldsymbol{\zeta}^{U\top} (\mathbf{u} - \mathbf{u}^U) = 0, \quad \boldsymbol{\zeta}^{L\top} (\mathbf{u}^L - \mathbf{u}) = 0 \quad (2.6)$$

$$\boldsymbol{\mu} \geq \mathbf{0}, \quad \boldsymbol{\zeta}^U \geq \mathbf{0}, \quad \boldsymbol{\zeta}^L \geq \mathbf{0} \quad (2.7)$$

$$\frac{\partial \mathcal{L}}{\partial \mathbf{u}} = \frac{\partial \Phi}{\partial \mathbf{u}} + \boldsymbol{\mu}^\top \frac{\partial \mathbf{G}}{\partial \mathbf{u}} + \boldsymbol{\zeta}^U - \boldsymbol{\zeta}^L = \mathbf{0}, \quad (2.8)$$

with the Lagrangian function defined as:

$$\mathcal{L}(\mathbf{u}, \boldsymbol{\mu}, \boldsymbol{\zeta}^U, \boldsymbol{\zeta}^L, \boldsymbol{\theta}) := \Phi + \boldsymbol{\mu}^\top \mathbf{G} + \boldsymbol{\zeta}^{U\top} (\mathbf{u} - \mathbf{u}^U) + \boldsymbol{\zeta}^{L\top} (\mathbf{u}^L - \mathbf{u}) \quad (2.9)$$

Proof. See, e.g., [7, Theorem 4.2.13]. \square

The necessary conditions of optimality (2.5) are referred to as the primal feasibility conditions, (2.6) as the complementarity slackness conditions, and (2.7, 2.8) as the dual feasibility conditions. Together these conditions are called the KKT conditions. Any point \mathbf{u}^* for which there exist Lagrange multipliers $\boldsymbol{\mu}^*, \boldsymbol{\zeta}^{U*}, \boldsymbol{\zeta}^{L*}$, such that $(\mathbf{u}^*, \boldsymbol{\mu}^*, \boldsymbol{\zeta}^{U*}, \boldsymbol{\zeta}^{L*})$ satisfies the KKT conditions is called a *KKT point*. The linear-independence constraint qualification implies that the associated Lagrange multipliers are determined uniquely at the KKT point \mathbf{u}^* .

The conditions in Theorem 2.1 are necessary conditions that must hold at each local minimum point for which LICQ holds. Nevertheless, these conditions may be satisfied by a point that is not a local minimum. Sufficient conditions for a KKT point to be a strict local minimum are given next.

2.1.3 Second-Order Sufficient Conditions

Let us denote the inequality constraints \mathbf{G} and the input bounds collectively as

$$\mathbf{z}(\mathbf{u}, \boldsymbol{\theta}) := \begin{bmatrix} \mathbf{G}(\mathbf{u}, \boldsymbol{\theta}) \\ \mathbf{u}^L - \mathbf{u} \\ \mathbf{u} - \mathbf{u}^U \end{bmatrix} \leq \mathbf{0}, \quad (2.10)$$

and the corresponding Lagrange multipliers as $\mathbf{v}^\top = (\boldsymbol{\mu}^\top, \boldsymbol{\zeta}^{\text{U}\top}, \boldsymbol{\zeta}^{\text{L}\top})$.

Theorem 2.2 (KKT Second-Order Sufficient Conditions)

Let \mathbf{u}^* be a KKT point for Problem (2.3), with the Lagrange multipliers $\mathbf{v}^{*\top} = (\boldsymbol{\mu}^{*\top}, \boldsymbol{\zeta}^{\text{U}\top}, \boldsymbol{\zeta}^{\text{L}\top})$. Let $\mathcal{A}_z := \{i : z_i(\mathbf{u}^*, \boldsymbol{\theta}) = 0, i = 1, \dots, n_z\}$, with $n_z = n_g + 2n_u$, and denote the strongly active constraints as $\mathcal{A}_z^+ := \{i \in \mathcal{A}_z : v_i^* > 0\}$ and the weakly active constraints as $\mathcal{A}_z^0 := \{i \in \mathcal{A}_z : v_i^* = 0\}$. Define the restricted Lagrangian function as

$$L(\mathbf{u}, \boldsymbol{\theta}) := \mathcal{L}(\mathbf{u}, \boldsymbol{\mu}^*, \boldsymbol{\zeta}^{\text{U}*}, \boldsymbol{\zeta}^{\text{L}*}, \boldsymbol{\theta}) = \Phi(\mathbf{u}, \boldsymbol{\theta}) + \sum_{i \in \mathcal{A}_z} v_i^* z_i(\mathbf{u}, \boldsymbol{\theta}), \quad (2.11)$$

and denote its Hessian at \mathbf{u}^* by

$$\frac{\partial^2 L}{\partial \mathbf{u}^2}(\mathbf{u}^*, \boldsymbol{\theta}) := \frac{\partial^2 \Phi}{\partial \mathbf{u}^2}(\mathbf{u}^*, \boldsymbol{\theta}) + \sum_{i \in \mathcal{A}_z} v_i^* \frac{\partial^2 z_i}{\partial \mathbf{u}^2}(\mathbf{u}^*, \boldsymbol{\theta}). \quad (2.12)$$

Define the cone

$$C = \left\{ \mathbf{d} \neq \mathbf{0} : \begin{aligned} \frac{\partial z_i}{\partial \mathbf{u}}(\mathbf{u}^*, \boldsymbol{\theta}) \mathbf{d} &= 0 && \text{for } i \in \mathcal{A}_z^+; \\ \frac{\partial z_i}{\partial \mathbf{u}}(\mathbf{u}^*, \boldsymbol{\theta}) \mathbf{d} &\leq 0 && \text{for } i \in \mathcal{A}_z^0 \end{aligned} \right\}$$

Then, if $\mathbf{d}^\top \frac{\partial^2 L}{\partial \mathbf{u}^2}(\mathbf{u}^*, \boldsymbol{\theta}) \mathbf{d} > 0$ for all $\mathbf{d} \in C$, \mathbf{u}^* is a strict local minimum for Problem (2.3).

Proof. See, e.g., [7, Theorem 4.4.2]. □

When the Lagrangian function $L(\mathbf{u}, \boldsymbol{\theta})$ displays a positive curvature at a KKT point \mathbf{u}^* along directions restricted to lie in the cone specified above, we claim that \mathbf{u}^* is a strict local minimum for Problem (2.3).

2.1.4 Reduced Gradient and Hessian

Let us assume that, at the optimum \mathbf{u}^* of Problem (2.3), there are n_g^a active inequality constraints \mathbf{G}^a , n^U inputs at their upper bound, and

n^L inputs at their lower bound. The active inequality constraints \mathbf{G}^a and the active input bounds are denoted collectively by \mathbf{z}^a . The null space of the Jacobian of the active constraints can be defined from the relation:

$$\frac{\partial \mathbf{z}^a}{\partial \mathbf{u}}(\mathbf{u}^*, \boldsymbol{\theta}) \mathbf{Z} = \mathbf{0}$$

where the columns of $\mathbf{Z} \in \mathbb{R}^{n_u \times n_s}$, with $n_s = n_u - n_g^a - n^U - n^L$, are a set of basis vectors for the null space of the active constraint Jacobian. The reduced gradient of the cost function, $\nabla_r \Phi \in \mathbb{R}^{1 \times n_s}$, is given by:

$$\nabla_r \Phi(\mathbf{u}^*, \boldsymbol{\theta}) = \frac{\partial \Phi}{\partial \mathbf{u}}(\mathbf{u}^*, \boldsymbol{\theta}) \mathbf{Z} \quad (2.13)$$

and the reduced Hessian of the cost function, $\nabla_r^2 \Phi \in \mathbb{R}^{n_s \times n_s}$, is given by [42]:

$$\nabla_r^2 \Phi(\mathbf{u}^*, \boldsymbol{\theta}) := \mathbf{Z}^\top \left[\frac{\partial^2 L}{\partial \mathbf{u}^2}(\mathbf{u}^*, \boldsymbol{\theta}) \right] \mathbf{Z} \quad (2.14)$$

2.1.5 Post-Optimal Sensitivity

Conditions under which a solution to Problem (2.3) exists and is unique for values of the adjustable parameters $\boldsymbol{\theta}$ in the neighborhood of some nominal values $\boldsymbol{\theta}_o$ are given in the following theorem.

Theorem 2.3 (Sensitivity analysis) *Let the functions Φ and G_i , $i = 1, \dots, n_g$, be twice continuously differentiable with respect to $(\mathbf{u}, \boldsymbol{\theta})$. Consider Problem (2.3) for the nominal parameter values $\boldsymbol{\theta}_o$, and let $\mathbf{u}^*(\boldsymbol{\theta}_o)$ be such that:*

1. *the second-order sufficient conditions for a local minimum of Problem (2.3) hold at $\mathbf{u}^*(\boldsymbol{\theta}_o)$, with the associated Lagrange multipliers $\boldsymbol{\mu}^*(\boldsymbol{\theta}_o)$, $\boldsymbol{\zeta}^{U^*}(\boldsymbol{\theta}_o)$ and $\boldsymbol{\zeta}^{L^*}(\boldsymbol{\theta}_o)$ (see Section 2.1.3);*
2. *the regularity LICQ condition (2.4) holds at $\mathbf{u}^*(\boldsymbol{\theta}_o)$;*
3. *the following strict complementarity conditions hold, $\mu_i^*(\boldsymbol{\theta}_o) > 0$ for each $i \in \mathcal{A}(\boldsymbol{\theta}_o)$, $\zeta_i^{L^*}(\boldsymbol{\theta}_o) > 0$ for each i such that $u_i^* = u_i^L$, and $\zeta_i^{U^*}(\boldsymbol{\theta}_o) > 0$ for each i such that $u_i^* = u_i^U$.*

Then, there is some $\eta > 0$ such that, for each $\boldsymbol{\theta} \in \mathcal{B}_\eta(\boldsymbol{\theta}_o)$ – a ball of radius η centered at $\boldsymbol{\theta}_o$, there exist unique continuously differentiable vector functions $\mathbf{u}^(\boldsymbol{\theta})$, $\boldsymbol{\mu}^*(\boldsymbol{\theta})$, $\boldsymbol{\zeta}^{L^*}(\boldsymbol{\theta})$ and $\boldsymbol{\zeta}^{U^*}(\boldsymbol{\theta})$ satisfying*

the second-order sufficient conditions for a local minimum of Problem (2.3). That is, $\mathbf{u}^*(\boldsymbol{\theta})$ is an isolated (local) minimizer of (2.3) with associated (unique) Lagrange multipliers $\boldsymbol{\mu}^*(\boldsymbol{\theta})$, $\boldsymbol{\zeta}^{\text{L}^*}(\boldsymbol{\theta})$, $\boldsymbol{\zeta}^{\text{U}^*}(\boldsymbol{\theta})$. Moreover, the set of active constraints remains identical, the Lagrange multipliers are such that $\mu_i^*(\boldsymbol{\theta}) > 0$, $\zeta_i^{\text{L}^*}(\boldsymbol{\theta}) > 0$ and $\zeta_i^{\text{U}^*}(\boldsymbol{\theta}) > 0$ for the active constraints, and the active constraints are linearly independent at $\mathbf{u}^*(\boldsymbol{\theta})$.

Proof. See, e.g., [30, Theorem 3.2.2]. □

The derivatives of the vector functions $\mathbf{u}^*(\boldsymbol{\theta})$, $\boldsymbol{\mu}^*(\boldsymbol{\theta})$, $\boldsymbol{\zeta}^{\text{L}^*}(\boldsymbol{\theta})$ and $\boldsymbol{\zeta}^{\text{U}^*}(\boldsymbol{\theta})$ at $\boldsymbol{\theta}_\circ$ are obtained by differentiating the first-order necessary conditions (2.6) and (2.8), which are to be satisfied for each $\boldsymbol{\theta} \in \mathcal{B}_\eta(\boldsymbol{\theta}_\circ)$:

$$\begin{bmatrix} \frac{\partial \mathbf{u}^*}{\partial \boldsymbol{\theta}}(\boldsymbol{\theta}_\circ) \\ \frac{\partial \boldsymbol{\mu}^*}{\partial \boldsymbol{\theta}}(\boldsymbol{\theta}_\circ) \\ \frac{\partial \boldsymbol{\zeta}^{\text{U}^*}}{\partial \boldsymbol{\theta}}(\boldsymbol{\theta}_\circ) \\ \frac{\partial \boldsymbol{\zeta}^{\text{L}^*}}{\partial \boldsymbol{\theta}}(\boldsymbol{\theta}_\circ) \end{bmatrix} = -\mathbf{M}(\boldsymbol{\theta}_\circ)^{-1} \mathbf{N}(\boldsymbol{\theta}_\circ). \quad (2.15)$$

The matrix $\mathbf{M}(\boldsymbol{\theta}_\circ) \in \mathbb{R}^{(n_g+3n_u) \times (n_g+3n_u)}$ stands for the Jacobian of (2.6, 2.8) with respect to $(\mathbf{u}, \boldsymbol{\mu}, \boldsymbol{\zeta}^{\text{U}}, \boldsymbol{\zeta}^{\text{L}})$ at $\boldsymbol{\theta}_\circ$,

$$\mathbf{M}(\boldsymbol{\theta}) := \begin{pmatrix} \frac{\partial^2 \mathcal{L}}{\partial \mathbf{u}^2} & \frac{\partial G_1}{\partial \mathbf{u}}^\top & \dots & \frac{\partial G_{n_g}}{\partial \mathbf{u}}^\top & \mathbf{I}_{n_u} & -\mathbf{I}_{n_u} \\ \mu_1 \frac{\partial \mathcal{L}}{\partial \mathbf{u}} & G_1 & & & & \\ \vdots & & \ddots & & & \\ \mu_{n_g} \frac{\partial \mathcal{L}}{\partial \mathbf{u}} & & & G_{n_g} & & \\ \text{diag}(\boldsymbol{\zeta}^{\text{U}}) & & & & \text{diag}(\mathbf{u} - \mathbf{u}^{\text{U}}) & \\ -\text{diag}(\boldsymbol{\zeta}^{\text{L}}) & & & & & \text{diag}(\mathbf{u}^{\text{L}} - \mathbf{u}) \end{pmatrix} \quad (2.16)$$

this is the so-called KKT matrix, the inverse of which is guaranteed to exist under the assumptions of Theorem 2.3. The matrix $\mathbf{N}(\boldsymbol{\theta}) \in \mathbb{R}^{(n_g+3n_u) \times n_\theta}$ stands for the Jacobian of (2.6, 2.8) with respect to $\boldsymbol{\theta}$ at $\boldsymbol{\theta}_\circ$,

$$\mathbf{N}(\boldsymbol{\theta}) := \begin{pmatrix} \frac{\partial^2 \mathcal{L}}{\partial \mathbf{u} \partial \boldsymbol{\theta}} \\ \mu_1 \frac{\partial G_1}{\partial \boldsymbol{\theta}} \\ \vdots \\ \mu_{n_g} \frac{\partial G_{n_g}}{\partial \boldsymbol{\theta}} \\ \mathbf{0}_{2n_u \times n_\theta} \end{pmatrix}. \quad (2.17)$$

2.2 Williams-Otto Reactor Problem

The reactor in the Williams-Otto plant [85], as modified by Roberts [72], will be used to illustrate the difficulties faced in RTO when the process model is subject to structural mismatch, and it will serve as a motivating example for the approaches studied in this thesis. Note that this reactor example has been used to illustrate model adequacy and RTO performance in numerous studies [34, 35, 89, 97].

2.2.1 Description and Model Equations

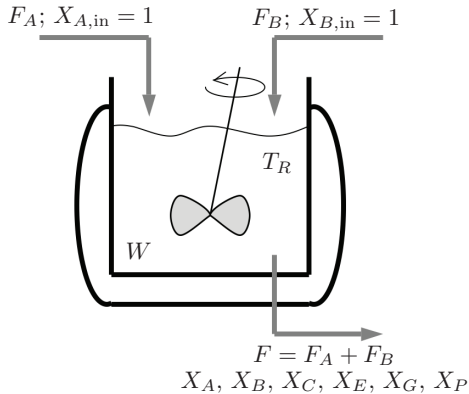
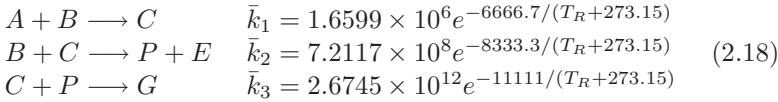


Fig. 2.1. Williams-Otto Reactor.

The reactor is illustrated in Figure 2.1. It consists of an ideal CSTR in which the following reactions occur:



where the reactants A and B are fed with the mass flow rates F_A and F_B , respectively. The desired products are P and E . C is an intermediate product and G is an undesired product. The product stream has the mass flow rate $F = F_A + F_B$. Operation is isothermal at the temperature T_R . The reactor mass holdup is $W = 2105$ kg.

Simulated reality: Plant equations

The steady-state model for the simulated reality results from the material balance equations:

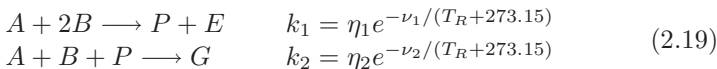
$$\begin{aligned}
0 &= F_A - (F_A + F_B)\bar{X}_A - W\bar{r}_1 \\
0 &= F_B - (F_A + F_B)\bar{X}_B - \frac{M_B}{M_A}W\bar{r}_1 - W\bar{r}_2 \\
0 &= -(F_A + F_B)\bar{X}_C + \frac{M_C}{M_A}W\bar{r}_1 - \frac{M_C}{M_B}W\bar{r}_2 - W\bar{r}_3 \\
0 &= -(F_A + F_B)\bar{X}_P + \frac{M_P}{M_B}W\bar{r}_2 - \frac{M_P}{M_C}W\bar{r}_3 \\
0 &= -(F_A + F_B)\bar{X}_G + \frac{M_G}{M_C}W\bar{r}_3 \\
\bar{X}_E &= \frac{M_E}{M_P}\bar{X}_P + \frac{M_E}{M_G}\bar{X}_G
\end{aligned}$$

with,

$$\bar{r}_1 = \bar{k}_1\bar{X}_A\bar{X}_B, \quad \bar{r}_2 = \bar{k}_2\bar{X}_B\bar{X}_C, \quad \bar{r}_3 = \bar{k}_3\bar{X}_C\bar{X}_P$$

The variables are: \bar{X}_i : mass fraction of species i , M_i : molecular weight of species i . W : mass holdup. k_j : kinetic coefficient of reaction j given in (2.18).

In this example, the reaction scheme (2.18) corresponds to the simulated reality. However, since it is assumed that the reaction scheme is not well understood, the following two reactions have been proposed to model the system [35]:



Two-reaction model: Model equations

The steady-state model for the two-reaction approximation results from the material balance equations:

$$\begin{aligned}
 0 &= F_A - (F_A + F_B)X_A - Wr_1 - Wr_2 \\
 0 &= F_B - (F_A + F_B)X_B - \frac{M_B}{M_A}2Wr_1 - \frac{M_B}{M_A}Wr_2 \\
 0 &= -(F_A + F_B)X_P + \frac{M_P}{M_A}Wr_1 - \frac{M_P}{M_A}Wr_2 \\
 0 &= -(F_A + F_B)X_E + \frac{M_E}{M_A}Wr_1 \\
 X_G &= \frac{M_G}{M_E}X_E + \frac{M_G}{M_P}X_P
 \end{aligned}$$

with,

$$r_1 = k_1 X_A X_B^2, \quad r_2 = k_2 X_A X_B X_P.$$

The kinetic coefficients k_1 and k_2 are given in (2.19).

By assuming $M_A = M_B = M_P$, all the molecular weight ratios that appear in the model equations are defined from the stoichiometry of the reactions.

Process Measurements

Forbes and Marlin [34] considered the process measurements $\mathbf{y}_p = (\bar{F}_B \ \bar{T}_R \ \bar{X}_A \ \bar{X}_B \ \bar{X}_C \ \bar{X}_E \ \bar{X}_G \ \bar{X}_P \ \bar{F})^\top$. However, since in this work measurement noise on the decision variables is not considered (the decision variables are assumed to be either setpoint values or known manipulated variables), the flow rates and reactor temperature are not considered as measurements. Also, the mass fraction of C is not included in the measurements since the presence of C is not assumed to be known, and it is not predicted by the model. Therefore, the measurement vector is chosen as $\mathbf{y}_p = (\bar{X}_A \ \bar{X}_B \ \bar{X}_E \ \bar{X}_G \ \bar{X}_P)^\top$.

Nominal model parameters

The four adjustable parameters $\boldsymbol{\theta} = (\eta_1 \ \nu_1 \ \eta_2 \ \nu_2)^\top$ are not identifiable from the process measurements \mathbf{y}_p . Two possible adjustable parameter sets were considered in [34]: in the first case, only the frequency

factors are updated ($\boldsymbol{\theta}_1 = (\eta_1 \ \eta_2)^\top$), while in the second case, only the activation energies are updated ($\boldsymbol{\theta}_2 = (\nu_1 \ \nu_2)^\top$). Both sets are identifiable from the process measurements. Here, the first set $\boldsymbol{\theta}_1 = (\eta_1 \ \eta_2)^\top$ is chosen. Typically, the fixed parameter values will be determined from detailed plant studies and/or engineering principles. In [34], the fixed parameters ν_1 and ν_2 were chosen such that the closed-loop RTO system converged to the same manipulated variable values for each adjustable set. In this work, the nominal values for the activation energies are the same as in [34], i.e., $\nu_1 = 8077.6$ K, and $\nu_2 = 12438.5$ K.

2.2.2 Optimization Problem

The objective is to maximize the operating profit, which is expressed as the cost difference between the product and reactant flowrates:

$$J(\mathbf{u}, \mathbf{y}_p) = 1143.38\bar{X}_P F + 25.92\bar{X}_E F - 76.23F_A - 114.34F_B, \quad (2.20)$$

which is equivalent to minimizing $\phi(\mathbf{u}, \mathbf{y}_p) = -J(\mathbf{u}, \mathbf{y}_p)$. The flowrate of reactant A (F_A) is fixed at 1.8275 kg/s. The flowrate of reactant B (F_B) and the reactor temperature (T_R) are the decision variables, thus $\mathbf{u} = (F_B, T_R)^\top$.

Plant optimum

The optimal solution for the plant (simulated reality) is presented in Table 2.1. The mass fractions obtained at the optimal solution are given in Table 2.2. The profit surface is strictly convex for the operating range considered, and is presented in Figure 2.2. The profit contours are given in Figure 2.3.

Table 2.1. Plant optimum

F_B^* [kg/s]	T_R^* [°C]	ϕ_{real}^* [\$/s]
4.787	89.70	190.98

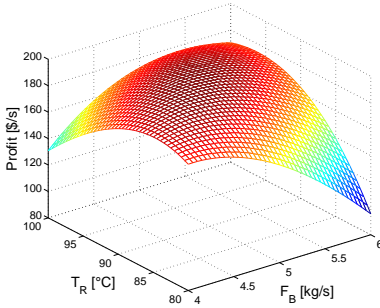


Fig. 2.2. Plant profit surface

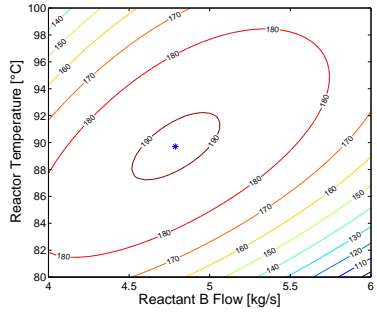


Fig. 2.3. Plant profit contours

Table 2.2. Plant mass fractions at the optimum

X_A^*	X_B^*	X_C^*	X_E^*	X_G^*	X_P^*
0.08746	0.38962	0.015306	0.10945	0.10754	0.29061

2.3 Classical Two-Step Approach to RTO

2.3.1 Principles

The classical two-step approach implies an iteration between parameter estimation and optimization [20, 46, 63]. The idea is to estimate repeatedly the model parameters θ of the nonlinear steady-state model, and to use the updated model in the model optimization to generate new inputs. This way, the model is expected to represent the plant at its current operating point more accurately. The parameter estimation and optimization problems at the RTO iteration k are given next:

Parameter Estimation Step

$$\theta_k^* \in \arg \min_{\theta} J^{\text{id}} \quad (2.21)$$

$$\text{with } J^{\text{id}} := [\mathbf{y}_p(\mathbf{u}_k) - \mathbf{y}(\mathbf{u}_k, \theta)]^T \mathbf{R} [\mathbf{y}_p(\mathbf{u}_k) - \mathbf{y}(\mathbf{u}_k, \theta)],$$

Optimization Step

$$\begin{aligned}
\mathbf{u}_{k+1}^* \in \arg \min_{\mathbf{u}} \quad & \Phi(\mathbf{u}, \boldsymbol{\theta}_k^*) := \phi(\mathbf{u}, \mathbf{y}(\mathbf{u}, \boldsymbol{\theta}_k^*)) & (2.22) \\
\text{s.t.} \quad & \mathbf{G}(\mathbf{u}, \boldsymbol{\theta}_k^*) := \mathbf{g}(\mathbf{u}, \mathbf{y}(\mathbf{u}, \boldsymbol{\theta}_k^*)) \leq \mathbf{0} \\
& \mathbf{u}^L \leq \mathbf{u} \leq \mathbf{u}^U,
\end{aligned}$$

Input Update

$$\begin{aligned}
\mathbf{u}_{k+1} &:= (\mathbf{I} - \mathbf{K})\mathbf{u}_k + \mathbf{K}\mathbf{u}_{k+1}^* & (2.23) \\
k &:= k + 1
\end{aligned}$$

In the parameter estimation problem (2.21), \mathbf{u}_k are the inputs applied to the plant at the RTO iteration k and $\mathbf{y}_p(\mathbf{u}_k)$ are the plant measurements obtained at steady state. The solution to the parameter estimation problem are the parameters $\boldsymbol{\theta}_k^*$ that are used to update the model in the next optimization problem. Given the solution to the optimization problem, \mathbf{u}_k^* , the first-order exponential filter (2.23) is typically employed, for which $\mathbf{K} \in \mathbf{R}^{n_u \times n_u}$ is a diagonal gain matrix with diagonal elements in the interval $(0, 1]$.

The local stability of the two-step RTO system at the plant optimum is discussed in [34]. The RTO system of Figure 2.4 is stable at the plant optimum \mathbf{u}_p^* for small perturbations in the inputs \mathbf{u} if:

$$\left\| \frac{d\mathbf{u}^*}{d\boldsymbol{\theta}} \bigg|_{\bar{\boldsymbol{\theta}}} \frac{d\boldsymbol{\theta}}{d\mathbf{y}_p} \bigg|_{\mathbf{y}_p(\mathbf{u}_p^*)} \frac{d\mathbf{y}_p}{d\mathbf{u}} \bigg|_{\mathbf{u}_p^*} \right\| < 1 \quad (2.24)$$

The first term in the product on the left-hand side of the inequality in condition (2.24) is the parametric sensitivity of the model-based optimization problem. The second term is the sensitivity of the adjustable parameters to changes in the measurements. The third term is the sensitivity of the process measurements (output variables) with respect to the inputs [34].

2.3.2 Model-Adequacy Criterion

The problem of model selection for closed-loop, real-time, model-based optimization is discussed in [34, 35]. A process model is said to be adequate for use in an RTO scheme if it is capable of producing a fixed

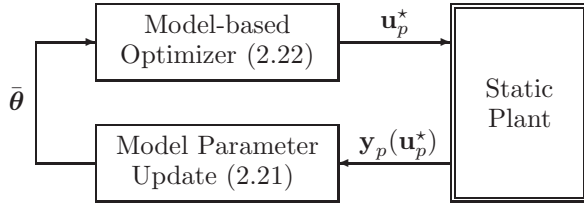


Fig. 2.4. The solution of Problem (2.1) \mathbf{u}_p^* seen as a fixed point of the two-step iterative RTO problem with the model update (2.21) and the model-based optimizer (2.22).

point for that RTO scheme at the plant optimum \mathbf{u}_p^* corresponding to the solution of Problem (2.1).

In other words, for the combined update/optimization system to be capable of predicting the optimal plant input variables from the process measurements, there must exist adjustable parameter values $\bar{\theta}$ for which the real-time optimizer generates the plant optimum values \mathbf{u}_p^* , while being the result of the model updating algorithm [34]. If such parameter values exist, the plant optimum \mathbf{u}_p^* is a fixed point (stationary point) for the two-step model-based RTO scheme (see Figure 2.4). Note that the parameter values $\bar{\theta}$ may not represent the true values of model parameters, especially in the case of structural plant-model mismatch for which the concept of ‘true values’ is not relevant. For the case where model update is performed by solving the parameter estimation problem (2.21), the following *model adequacy criterion* has been proposed [34].

Criterion 2.1 (Model Adequacy for the Classical Two-Step RTO Scheme) *Let \mathbf{u}_p^* be the unique plant optimum and assume that there exists (at least) one set of values $\bar{\theta}$ for the model parameters such that:*

$$G_i(\mathbf{u}_p^*, \bar{\boldsymbol{\theta}}) = 0, \quad i \in \mathcal{A}(\mathbf{u}_p^*) \quad (2.25)$$

$$G_i(\mathbf{u}_p^*, \bar{\boldsymbol{\theta}}) < 0, \quad i \notin \mathcal{A}(\mathbf{u}_p^*) \quad (2.26)$$

$$\nabla_r \Phi(\mathbf{u}_p^*, \bar{\boldsymbol{\theta}}) = \mathbf{0} \quad (2.27)$$

$$\nabla_r^2 \Phi(\mathbf{u}_p^*, \bar{\boldsymbol{\theta}}) \succ \mathbf{0} \quad (\text{positive definite}) \quad (2.28)$$

$$\frac{\partial J^{\text{id}}}{\partial \boldsymbol{\theta}}(\mathbf{y}_p(\mathbf{u}_p^*), \mathbf{y}(\mathbf{u}_p^*, \bar{\boldsymbol{\theta}})) = \mathbf{0} \quad (2.29)$$

$$\frac{\partial^2 J^{\text{id}}}{\partial \boldsymbol{\theta}^2}(\mathbf{y}_p(\mathbf{u}_p^*), \mathbf{y}(\mathbf{u}_p^*, \bar{\boldsymbol{\theta}})) \succ \mathbf{0} \quad (\text{positive definite}), \quad (2.30)$$

where $\nabla_r \Phi$ and $\nabla_r^2 \Phi$ are the reduced gradient and reduced Hessian of the cost function, respectively (see Subsection 2.1.4); and J^{id} stands for the performance index in the (unconstrained) parameter estimation problem. Then, the process model is adequate for use in the two-step RTO scheme given by (2.21) and (2.22).

Notice that both the active inequality constraints and the active input bounds are taken into account in the definition of the reduced gradient and the reduced Hessian of the cost function, given by equations (2.13) and (2.14), respectively.

Conditions (2.25)–(2.28) represent sufficient conditions for \mathbf{u}_p^* to be a strict local minimum of (2.22) with the parameters chosen as $\bar{\boldsymbol{\theta}}$, whereas Conditions (2.29)–(2.30) are sufficient for $\bar{\boldsymbol{\theta}}$ to be a strict local minimum of the estimation problem at \mathbf{u}_p^* . Hence, if all these conditions hold, the plant optimum \mathbf{u}_p^* corresponds to a (local) model optimum for $\boldsymbol{\theta} = \bar{\boldsymbol{\theta}}$, and Conditions (2.25)–(2.30) are *sufficient* for model adequacy. However, these conditions are *not necessary* for model adequacy. It may indeed be that \mathbf{u}_p^* corresponds to the model optimum (for $\boldsymbol{\theta} = \bar{\boldsymbol{\theta}}$) – i.e., the model is adequate –, but Conditions (2.25)–(2.30) are not met. One such situation is when the reduced Hessian is only semi-definite positive, for which (2.28) does not hold.

Noting that the equalities (2.29) alone yield n_θ conditions, it becomes clear that the full set of adequacy conditions (2.25)–(2.30) is over-specified. In other words, these model adequacy conditions are often not satisfied in the presence of plant-model mismatch.

Also, since \mathbf{u}_p^* is not known, the criterion cannot be used to select between candidate models and to select among all the model parameters which ones to adjust.

Example 2.1 Consider the plant $y_p = \Phi_p = (u - 1)^2$. It is easy to verify that the model $y = \Phi = 2(u - \theta)^2$ satisfies the adequacy conditions although it is structurally incorrect (see Definition 2.1). However, the model $y = \Phi = \theta + u^2$, which is also structurally incorrect, does not satisfy the adequacy conditions.

2.3.3 Numerical Illustration

Classical Two-Step Approach to RTO

The classical two-step approach is applied (without measurement noise) to the Williams-Otto reactor example described in Section 2.2. The trajectory of the RTO iterates obtained without filtering ($\mathbf{K} = \mathbf{I}$) when starting from three different initial operating points are shown in Figure 2.5. In all cases, the algorithm converges to the point $\mathbf{u}_\infty = (4.805, 84.0)^\top$ which is different from the true optimum $\mathbf{u}_p^* = (4.787, 89.7)^\top$. For the initial points corresponding to $\mathbf{u}_0 = (3.000, 70.0)^\top$ and $\mathbf{u}_0 = \mathbf{u}_p^*$, the evolution of the input variables with the RTO period is given in Figure 2.6. The evolution of the estimated model parameters is given in Figure 2.7 and that of the plant profit in Figure 2.8. At the stationary point reached upon convergence, the sum of squared error is $J^{\text{id}} = 9 \times 10^{-5}$ with $\mathbf{R} = \mathbf{I}_5$.

This example serves to illustrate the well-known fact that, in the presence of structural plant-model mismatch, the classical two-step approach of repeated parameter estimation and optimization, if convergent, may not improve plant operation. The correct plant optimum will not be achieved unless, in addition to matching model outputs with corresponding real process outputs, the derivatives of the constraints and the cost function with respect to the inputs also match at the final converged point [8, 27, 28, 74].

Model Adequacy

Since there are no active constraints at the plant optimum, the reduced gradient of the cost function corresponds to the full gradient. In order to search for adequate parameter values for $\boldsymbol{\theta} = [\eta_1 \ \eta_2]^\top$, i.e. parameter values capable of computing the plant optimum, the following parameter estimation problem is studied at the plant optimum:

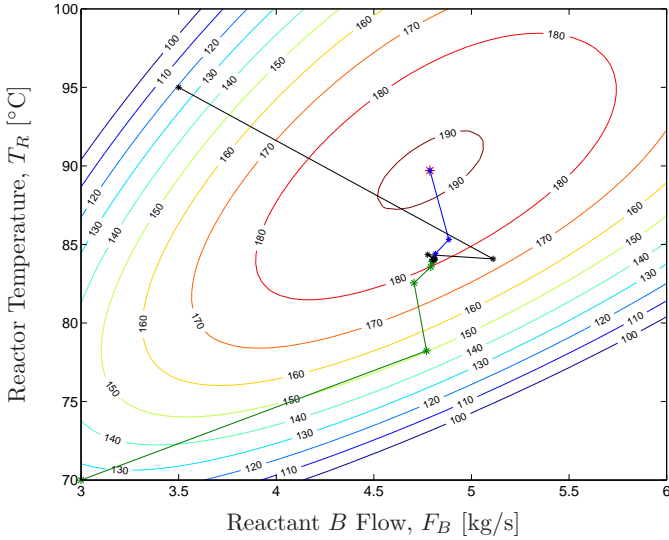


Fig. 2.5. Classical two-step approach. RTO trajectory.

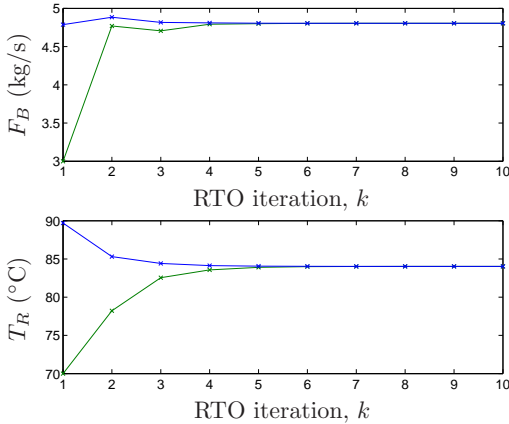


Fig. 2.6. Classical two-step approach. Evolution of the input variables from two distinct initial guesses.

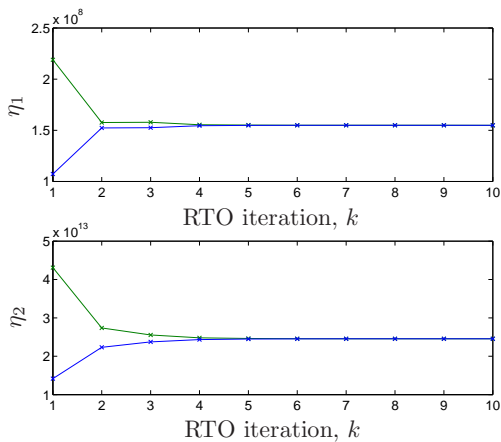


Fig. 2.7. Classical two-step approach. Evolution of the corresponding adjustable parameters.

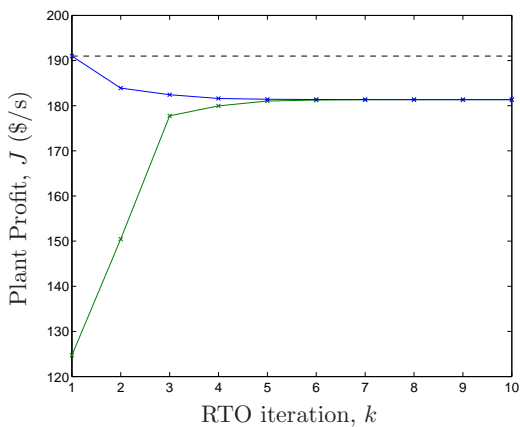


Fig. 2.8. Classical two-step approach. Evolution of the corresponding process profit.

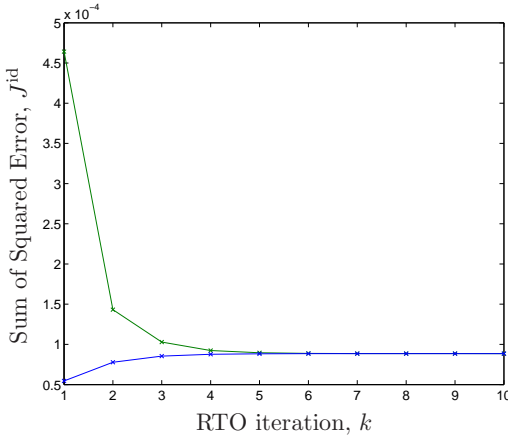


Fig. 2.9. Classical two-step approach. Evolution of the corresponding sum of squared error.

$$\begin{aligned} \min_{\theta} \quad & J^{\text{adeq}} := \frac{\partial \Phi}{\partial \mathbf{u}}(\mathbf{u}_p^*, \theta) \frac{\partial \Phi}{\partial \mathbf{u}}^{\top}(\mathbf{u}_p^*, \theta) \quad (2.31) \\ \text{s.t.} \quad & \text{Model equations} \end{aligned}$$

Two local minima are found for Problem (2.31), which are indicated by the letters **a** and **b** in Figure 2.11. These two local minima satisfy the optimality conditions (2.27) and (2.28), since J^{adeq} in Problem (2.31) vanishes at both points.

Table 2.3 presents the parameter values **a** and **b** as well as the solution $[F_B^* \ T_R^*]$ to the model-based optimization problem. For both **a** and **b**, the solution corresponds to the true plant optimum. Table 2.4 gives the mass fractions for the plant and the two-reaction model evaluated at the plant optimum for the parameter values **a** and **b**.

Table 2.3. Adequate parameters and corresponding optimal solution.

	η_1	η_2	J^{adeq}	F_B^*	T_R^*	ϕ^*
a	1.0728×10^8	1.4155×10^{13}	2.78×10^{-13}	4.787	89.70	190.98
b	1.1965×10^{10}	3.5752×10^{13}	5.66×10^{-11}	4.787	89.70	190.98

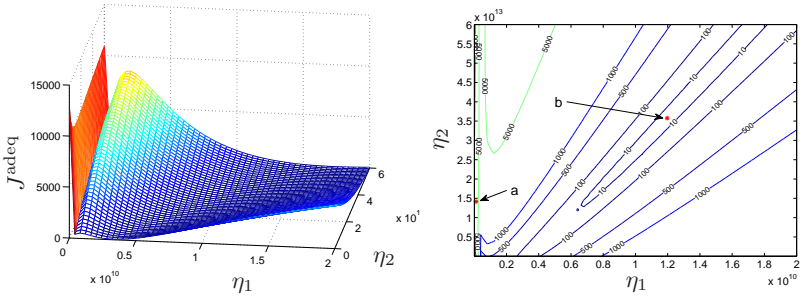


Fig. 2.10. J^{adeq} surface for Prob- **Fig. 2.11.** J^{adeq} contours for Problem 2.31.

Table 2.4. Mass fractions for the plant and the model evaluated at the plant optimum for the parameter values **a** and **b**.

	X_A^*	X_B^*	X_E^*	X_G^*	X_P^*	J^{id}
Plant	0.08746	0.38962	0.10945	0.10754	0.29061	
a	0.10857	0.41735	0.10967	0.27736	0.08702	0.06257
b	0.00847	0.19433	0.25541	0.52321	0.01859	0.22472

The sum of squared error of the mass fractions $J^{\text{id}} = (\mathbf{y}_p - \mathbf{y})^\top (\mathbf{y}_p - \mathbf{y})$ is also given in Table 2.4. Notice that the parameter values corresponding to both **a** and **b** yield the true plant optimum by numerical optimization based on the model. However, at the plant optimum, the adjustable parameter values corresponding to **a** predict more closely the plant outputs (mass fractions) than those corresponding to **b**. Therefore, it appears that solution **a** is physically more meaningful than solution **b**. However, neither **a** nor **b** is generated by the parameter estimation problem, that is, Condition (2.29) is not satisfied for these points, and therefore the two-reaction model is not adequate for use in the two-step RTO scheme given by (2.21) and (2.22).

2.4 ISOPE and Extensions

In response to the deficiencies of the classical two-step approach, a modified two-step approach known as *Integrated System Optimization and Parameter Estimation (ISOPE)* was proposed by Roberts

and co-workers in the late 70's [72, 74]. Since then, several extensions and variants of ISOPE have been proposed [13, 48, 73]. ISOPE algorithms include a parameter estimation problem and a modified optimization problem, which incorporates plant gradient information into a gradient-modification term that is added to the cost function. This modification term is derived from the first-order NCO (see Subsection 2.1.2) in such a way that, upon convergence, the iterates reach a KKT point for the plant.

2.4.1 ISOPE

In the ISOPE literature, an important distinction is made between optimization problems with and without inequality constraints that depend on the output \mathbf{y} , i.e., process-dependent constraints of the form $\mathbf{g}(\mathbf{u}, \mathbf{y}) \leq \mathbf{0}$ [13, 73]. For instance, the original ISOPE formulation, which does not consider the constraints, modifies only the gradient of the cost function [72, 74]. Optimality and convergence analysis for the original ISOPE algorithm have been studied [11]. ISOPE has later been extended to include process-dependent constraints [10, 94]. However, instead of including additional modifiers for the constraint functions as recently proposed by Gao and Engell [37], a Lagrangian modifier was introduced in the cost function. Also, to the author's best knowledge, no convergence analysis has been provided for the extended algorithm to date.

The original ISOPE algorithm, which does not consider process-dependent constraints in the optimization problem, reads:

$$\begin{aligned} \mathbf{u}_{k+1}^* &\in \arg \min_{\mathbf{u}} \phi(\mathbf{u}, \mathbf{y}(\mathbf{u}, \boldsymbol{\theta}_k)) + \boldsymbol{\lambda}_k^\top \mathbf{u} \\ \text{s.t.} \quad &\mathbf{u}^L \leq \mathbf{u} \leq \mathbf{u}^U. \end{aligned} \quad (2.32)$$

At the k th RTO iteration with the input \mathbf{u}_k , a parameter estimation problem is solved yielding the updated parameter values $\boldsymbol{\theta}_k$. This problem is solved under the additional constraint

$$\mathbf{y}(\mathbf{u}_k, \boldsymbol{\theta}_k) = \mathbf{y}_p(\mathbf{u}_k). \quad (2.33)$$

Then, assuming that the plant gradient $\frac{\partial \mathbf{y}_p}{\partial \mathbf{u}}(\mathbf{u}_k)$ is available, the ISOPE modifier $\boldsymbol{\lambda}_k$ is calculated as:

$$\boldsymbol{\lambda}_k^\top := \frac{\partial \phi}{\partial \mathbf{y}}(\mathbf{u}_k, \mathbf{y}(\mathbf{u}_k, \boldsymbol{\theta}_k)) \left[\frac{\partial \mathbf{y}_p}{\partial \mathbf{u}}(\mathbf{u}_k) - \frac{\partial \mathbf{y}}{\partial \mathbf{u}}(\mathbf{u}_k, \boldsymbol{\theta}_k) \right]. \quad (2.34)$$

The output-matching requirement (2.33) represents a model-qualification condition that is found throughout the ISOPE literature [11, 13, 74, 94]. The new operating point is determined by filtering the inputs using a first-order exponential filter, similar to (2.23).

2.4.2 ISOPE with Model Shift

More recently, it has been found that the condition (2.33) can be satisfied without updating the model parameters $\boldsymbol{\theta}$ [82]. This variant introduces the model shift term $\mathbf{a}_k := \mathbf{y}_p(\mathbf{u}_k) - \mathbf{y}(\mathbf{u}_k, \boldsymbol{\theta})$ in the modified optimization problem, so that (2.32) becomes:

$$\begin{aligned} \mathbf{u}_{k+1}^* &\in \arg \min_{\mathbf{u}} \phi(\mathbf{u}, \mathbf{y}(\mathbf{u}, \boldsymbol{\theta}) + \mathbf{a}_k) + \boldsymbol{\lambda}_k^\top \mathbf{u} & (2.35) \\ \text{s.t. } &\mathbf{u}^L \leq \mathbf{u} \leq \mathbf{u}^U, \end{aligned}$$

with

$$\boldsymbol{\lambda}_k^\top := \frac{\partial \phi}{\partial \mathbf{y}}(\mathbf{u}_k, \mathbf{y}_k + \mathbf{a}_k) \left[\frac{\partial \mathbf{y}_p}{\partial \mathbf{u}}(\mathbf{u}_k) - \frac{\partial \mathbf{y}}{\partial \mathbf{u}}(\mathbf{u}_k, \boldsymbol{\theta}_k) \right]. \quad (2.36)$$

Note that the name ISOPE becomes inappropriate for this variant, since it eliminates the need for estimating the model parameters.

2.4.3 ISOPE with FFD Approach

Perhaps the major difficulty with the ISOPE approach lies in the fact that the gradients of the plant outputs with respect to the input variables, also called experimental gradients $\frac{\partial \mathbf{y}_p}{\partial \mathbf{u}}$, must be available. A brief discussion on the methods for estimating the experimental gradients from plant measurements was given in Subsection 1.2.6.

A straightforward approach consists in perturbing each input individually around the current operating point to get an estimate of the corresponding gradient element. This is the case, e.g., when forward finite differencing (FFD) is applied at each RTO iteration. In the FFD approach, an estimator of the partial derivative $\frac{\partial y_p}{\partial u_j}(\mathbf{u}_k)$, $j = 1, \dots, n_u$, at the RTO iteration k , is obtained as

$$\hat{\beta}_j(h) = [y_p(\mathbf{u}_k + h\mathbf{e}_j) - y_p(\mathbf{u}_k)]/h, \quad h > 0, \quad (2.37)$$

where h is the step size and \mathbf{e}_j is the j th unit vector.

This approach requires n_u perturbations to be carried out at each RTO iteration, and for each perturbation a new steady-state must be attained. If during the gradient evaluation, an important exogenous perturbation or a change in operating conditions takes place, the gradients estimated by this approach may become meaningless.

2.4.4 Dual ISOPE

The idea behind dual ISOPE is to estimate the gradients based on the current and past operating points instead of perturbing each input individually at each iteration [12, 13]. This way, the total number of input changes required by the algorithm is expected to be less. The key issue is the ability to estimate experimental gradients reliably while updating the operating point. The input perturbations required in order to estimate the gradients might be suboptimal, and even infeasible in the presence of constraints. Indeed, there are two conflicting objectives: the “primal objective” consists in solving the optimization problem, while the “dual objective” aims at estimating accurate experimental gradients. These conflicting tasks can be accommodated by adding a constraint in the optimization problem so as to ensure sufficient information in the measurements and guarantee gradient accuracy. In [12, 13], a constraint that prevents ill-conditioning in gradient computation has been proposed.

Given the current operating point, \mathbf{u}_k , the n_u past operating points, $\mathbf{u}_{k-1}, \dots, \mathbf{u}_{k-n_u}$, and the corresponding measured plant output values, $\mathbf{y}_p(\mathbf{u}_k), \mathbf{y}_p(\mathbf{u}_{k-1}), \dots, \mathbf{y}_p(\mathbf{u}_{k-n_u})$, an estimate of the gradient $\frac{\partial \mathbf{y}_p}{\partial \mathbf{u}}(\mathbf{u}_k)$ can be obtained as follows [13]:

$$\hat{\boldsymbol{\beta}}_k := \mathbf{Y}(\mathbf{u}_k) \mathcal{U}^{-1}(\mathbf{u}_k) \quad (2.38)$$

with

$$\mathcal{U}(\mathbf{u}_k) := [\mathbf{u}_k - \mathbf{u}_{k-1} \ \dots \ \mathbf{u}_k - \mathbf{u}_{k-n_u}] \in \mathbb{R}^{n_u \times n_u} \quad (2.39)$$

$$\mathbf{Y}(\mathbf{u}_k) := [\mathbf{y}_p(\mathbf{u}_k) - \mathbf{y}_p(\mathbf{u}_{k-1}) \ \dots \ \mathbf{y}_p(\mathbf{u}_k) - \mathbf{y}_p(\mathbf{u}_{k-n_u})] \in \mathbb{R}^{n_y \times n_u} \quad (2.40)$$

Estimating the gradient from (2.38) requires the matrix $\mathcal{U}(\mathbf{u}_k)$ to be non-singular. Furthermore, since the measurements \mathbf{y}_p are corrupted by measurement noise, it has been proposed in [12] to ensure that matrix $\mathcal{U}(\mathbf{u}_k)$ is well-conditioned. In dual ISOPE algorithms, good

conditioning is achieved by introducing a lower bound on the inverse condition number of $\mathcal{U}(\mathbf{u}_k)$:

$$\kappa^{-1}(\mathbf{u}_k, \mathbf{u}_{k-1}, \dots, \mathbf{u}_{k-n_u}) := \frac{\sigma_{\min}(\mathcal{U}(\mathbf{u}_k))}{\sigma_{\max}(\mathcal{U}(\mathbf{u}_k))} \geq \varphi, \quad (2.41)$$

where $\kappa(\mathbf{u}_k, \mathbf{u}_{k-1}, \dots, \mathbf{u}_{k-n_u})$ is the condition number of $\mathcal{U}(\mathbf{u}_k)$.

The dual ISOPE algorithm can be summarized as follows [13]:

1. Select an initial point \mathbf{u}_0 and n_u additional points $\mathbf{u}_{-n_u}, \mathbf{u}_{-n_u+2}, \dots, \mathbf{u}_{-1}$ such that the matrix $\mathcal{U}(\mathbf{u}_0)$ is sufficiently well conditioned. Initialize the parameters of the algorithm, i.e. \mathbf{K} , φ and the parameter ϵ corresponding to the termination criterion $\|\mathbf{u}_{k+1} - \mathbf{u}_k\| \leq \epsilon$. Apply the points $\mathbf{u}_{-n_u}, \mathbf{u}_{-n_u+2}, \dots, \mathbf{u}_{-1}$ to the plant and measure the plant outputs $\mathbf{y}_p(\mathbf{u}_j), \mathbf{y}_p(\mathbf{u}_{-n_u+2}), \dots, \mathbf{y}_p(\mathbf{u}_{-1})$ at steady-state. Set $k = 0$.
2. Apply \mathbf{u}_k to the plant and measure $\mathbf{y}_p(\mathbf{u}_k)$ at steady-state operation. Obtain an estimate of the plant gradient $\hat{\beta}_k$ according to (2.38).
3. Update the model parameters θ_k by solving a parameter estimation problem with the constraint of matching the plant and model outputs at \mathbf{u}_k (see output-matching requirement (2.33)).
4. Compute the ISOPE modifier λ_k as:

$$\lambda_k^\top := \frac{\partial \phi}{\partial \mathbf{y}}(\mathbf{u}_k, \mathbf{y}(\mathbf{u}_k, \theta_k)) \left[\hat{\beta}_k - \frac{\partial \mathbf{y}}{\partial \mathbf{u}}(\mathbf{u}_k, \theta_k) \right], \quad (2.42)$$

and solve the following modified optimization problem including the conditioning constraint:

$$\begin{aligned} \mathbf{u}_{k+1}^* &\in \arg \min_{\mathbf{u}} \phi(\mathbf{u}, \mathbf{y}(\mathbf{u}, \theta_k)) + \lambda_k^\top \mathbf{u} & (2.43) \\ \text{s.t.} \quad &\kappa^{-1}(\bar{\mathbf{u}}, \mathbf{u}_k, \dots, \mathbf{u}_{k-n_u+1}) \geq \varphi, \\ &\bar{\mathbf{u}} = (\mathbf{I} - \mathbf{K})\mathbf{u}_k + \mathbf{K}\mathbf{u}, \\ &\mathbf{u}^L \leq \mathbf{u} \leq \mathbf{u}^U, \end{aligned}$$

5. Set $\mathbf{u}_{k+1} := (\mathbf{I} - \mathbf{K})\mathbf{u}_k + \mathbf{K}\mathbf{u}_{k+1}^*$. If $\|\mathbf{u}_{k+1} - \mathbf{u}_k\| \leq \epsilon$, terminate. Else, set $k := k + 1$ and return to Step 2.

Notice that in (2.43) $\bar{\mathbf{u}}$ corresponds to the filtered value of the input.

Example 2.2 Consider a two-dimensional optimization problem ($n_u = 2$). The left-side plot of Figure 2.12 uses the past operating points $\mathbf{u}_k = [0 \ -0.5]^\top$ and $\mathbf{u}_{k-1} = [0 \ 0.5]^\top$, while the right-side plot uses the past operating points $\mathbf{u}_k = [0 \ -0.1]^\top$ and $\mathbf{u}_{k-1} = [0 \ 0.1]^\top$. The conditioning constraint $\kappa^{-1}(\mathbf{u}, \mathbf{u}_k, \mathbf{u}_{k-1})$ can be evaluated in terms of the location of the new operating point $\mathbf{u} = [u_1 \ u_2]^\top$. The contour levels corresponding to values of κ^{-1} of 0.1, 0.2, 0.3 and 0.4 are represented.

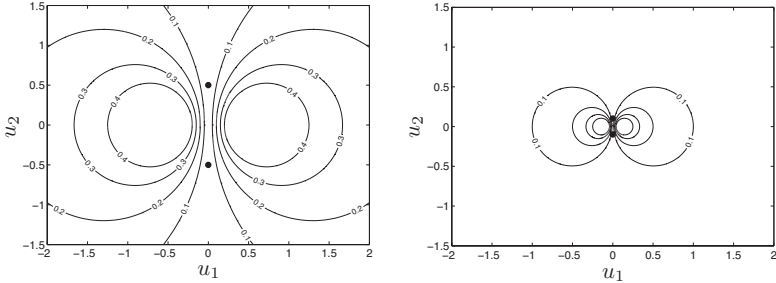


Fig. 2.12. Contour maps for the conditioning constraint (2.41) for two cases of previous points (more distant on the left and closer on the right).

The conditioning set $\mathcal{D}(\mathbf{u}, \mathbf{u}_k, \mathbf{u}_{k-1})$ is defined as:

$$\mathcal{D}(\mathbf{u}, \mathbf{u}_k, \mathbf{u}_{k-1}) := \left\{ \mathbf{u} \in \mathbb{R}^{n_u} : \frac{\sigma_{\min}(\mathcal{U}(\mathbf{u}))}{\sigma_{\max}(\mathcal{U}(\mathbf{u}))} \geq \varphi \right\},$$

The geometric characterisation of the set \mathcal{D} has been analyzed in [13, 81] for the two-dimensional case. It consists of two discs of the same radius r , located symmetrically with respect to the line defined by the points \mathbf{u}_k and \mathbf{u}_{k-1} . The discs are centered at a distance h_d from the centroid of the points \mathbf{u}_k and \mathbf{u}_{k-1} , in a direction orthogonal to the line defined by \mathbf{u}_k and \mathbf{u}_{k-1} , with

$$r = \frac{1 - \varphi^2}{4\varphi} \|\mathbf{u}_k - \mathbf{u}_{k-1}\|$$

$$h_d = \frac{1 + \varphi^2}{4\varphi} \|\mathbf{u}_k - \mathbf{u}_{k-1}\|$$

Finally, not mentioned in [13, 81] but of the greatest importance is the scaling of the input variables, since the conditioning set \mathcal{D} will completely depend on this scaling.

Real-Time Optimization via Constraint Adaptation

In the presence of uncertainty, the constraints predicted by the model do not quite match the actual constraints of the plant. This chapter investigates a so-called constraint-adaptation algorithm, which uses measurements for adapting the process constraints while relying on a fixed process model for satisfying the remaining parts of the NCO. The idea behind constraint adaptation is to use measurements of the constraints for modifying the constraints in the optimization problem between successive RTO periods so as to track the actual constraint values. This way, constraint adaptation guarantees that a feasible, yet suboptimal, operating point is attained upon convergence. Suboptimality will depend on how well the process model predicts the gradient of the cost function and of the active constraints. In addition to the constraint modifiers used in constraint adaptation, the full modifier-adaptation scheme studied in Chapter 4 also makes use of modifiers for the cost and constraint gradients in order to enforce optimality. Despite being a subproblem of modifier-adaptation, constraint adaptation deserves to be studied on its own because of its simplicity, as it does not require that the gradient of the cost and constraint functions be estimated experimentally, and also because it can provide a satisfactory solution for a large number of optimization problems for which most of the optimization potential lies in keeping the correct set of constraints active. Although constraint adaptation converges to a feasible operating point, the constraints can be violated during steady-state operation at the operating points produced by the algorithm prior to convergence, or during the dynamic response between successive steady-state operating points. A convenient way to avoid these infeasibilities is by means

of model predictive control (MPC) tailored to control the constraints. Section 3.3 presents an approach for integrating constraint adaptation with MPC. The applicability and suitability of constraint adaptation is demonstrated through the case study of a solid-oxide fuel cell (SOFC) system in Subsection 3.3.3.

3.1 Variational Analysis of NCO in the Presence of Parametric Uncertainty

Satisfying the NCO is necessary for process operation to be feasible and achieve optimal performance. More precisely, failure to meet any of the primal feasibility conditions (2.5) leads to infeasible operation, whereas violating the complementarity and dual feasibility conditions (2.6-2.8) results in suboptimal operation. That is, a very desirable property for a RTO scheme is that the iterates converge to a point at which the NCO of the real process are satisfied. Regarding fixed-model RTO methods, the algorithm described in [37], provided it converges, can be shown to attain such points. However, for RTO schemes to satisfy certain NCO, it is necessary that the gradients of the cost and constraint functions be available, which in turn requires that these gradients be estimated experimentally from the available measurements.

In this section, we conduct a variational analysis of the NCO in the presence of uncertainty. The underlying idea is to quantify how deviations of the parameters $\boldsymbol{\theta}$ from their nominal values $\boldsymbol{\theta}_o$ affect the NCO and, through them, the performance of the process. Said differently, the objective is to determine which parts of the NCO influence the process performance the most, and which parts can be dropped without much impact in terms of performance. For simplicity, and without loss of generality, the *bound constraints* $\mathbf{u}^L \leq \mathbf{u} \leq \mathbf{u}^U$ shall be dropped, and only the general inequality constraints $\mathbf{G}(\mathbf{u}, \boldsymbol{\theta}) \leq \mathbf{0}$ shall be considered for this analysis.¹

In this section, a minimizing solution of Problem (2.3) will be denoted by $\mathbf{u}_{\boldsymbol{\theta}}^*$ when we need to emphasize the dependency on $\boldsymbol{\theta}$. Also, we shall assume, without loss of generality, that *all the constraints* $\mathbf{G}(\cdot, \boldsymbol{\theta}_o)$ are active at $\mathbf{u}_{\boldsymbol{\theta}_o}^*$, since an inactive constraint at $\mathbf{u}_{\boldsymbol{\theta}_o}^*$ can always be removed from Problem (2.3). The number of active constraints

¹ Alternatively, just replace $\mathbf{G}(\mathbf{u}, \boldsymbol{\theta}) \leq \mathbf{0}$ by the general constraints $\mathbf{z}(\mathbf{u}, \boldsymbol{\theta}) \leq \mathbf{0}$ defined in (2.10).

at an optimal solution point being less than the number of input variables, there exist directions in the input space along which taking an infinitesimal step from an optimal solution point does not modify the active constraints. A characterization of these directions is given in the next definition.

Definition 3.1 (Constraint-Seeking and Sensitivity-Seeking Directions) *If the assumptions of Theorem 2.3 are met, the Jacobian matrix $\frac{\partial \mathbf{G}}{\partial \mathbf{u}}$ is full rank at $(\mathbf{u}_{\theta_o}^*, \theta_o)$. Singular value decomposition of $\frac{\partial \mathbf{G}}{\partial \mathbf{u}}$ gives*

$$\frac{\partial \mathbf{G}}{\partial \mathbf{u}}(\mathbf{u}_{\theta_o}^*, \theta_o) := \mathbf{U} \Sigma \mathbf{V}^T,$$

where Σ is a $(n_g \times n_u)$ matrix of the form

$$\Sigma = \begin{pmatrix} \sigma_1 & & 0 & \cdots & 0 \\ & \ddots & & & \\ & & \sigma_{n_g} & & 0 \\ & & & 0 & \cdots & 0 \end{pmatrix} := [\Sigma^c \mathbf{0}];$$

\mathbf{U} is an $(n_g \times n_g)$ orthonormal matrix; and $\mathbf{V} := [\mathbf{V}^c \mathbf{V}^s]$ is an $(n_u \times n_u)$ orthonormal matrix. The n_g columns of \mathbf{V}^c define the constraint-seeking directions, while the $(n_u - n_g)$ columns of \mathbf{V}^s define the sensitivity-seeking directions.

Clearly, infinitesimal moves along a constraint-seeking direction away from an optimal solution point modify the active constraint values, whereas infinitesimal moves along a sensitivity-seeking direction leave the active constraints unchanged. The subsequent theorem quantifies the variation of the optimal inputs, in both the constraint- and sensitivity-seeking directions, which is induced by a deviation of the model parameters from their nominal values.

Theorem 3.1 (Input Variations in the Constraint-Seeking and Sensitivity-Seeking Directions) *Let the assumptions in Theorem 2.3 be met and η be chosen such that Theorem 2.3 applies. Let $\delta \mathbf{u}^* := \mathbf{u}_{\theta}^* - \mathbf{u}_{\theta_o}^*$ denote the variation of the optimal inputs induced by the variation $\delta \theta := \theta - \theta_o$, with $\theta \in \mathcal{B}_\eta(\theta_o)$, and consider the variations $\delta \mathbf{u}^c$ and $\delta \mathbf{u}^s$ of the optimal inputs in the constraint-seeking and sensitivity-seeking directions, respectively,*

$$\begin{bmatrix} \delta \mathbf{u}^c \\ \delta \mathbf{u}^s \end{bmatrix} := [\mathbf{V}^c \mathbf{V}^s]^T \delta \mathbf{u}^*.$$

Then, the first-order approximations of $\delta \mathbf{u}^c$ and $\delta \mathbf{u}^s$ are given by:

$$\delta \mathbf{u}^c = - \left(\frac{\partial \mathbf{G}}{\partial \mathbf{u}} \mathbf{V}^c \right)^{-1} \frac{\partial \mathbf{G}}{\partial \boldsymbol{\theta}} \delta \boldsymbol{\theta} + o(\|\delta \boldsymbol{\theta}\|) \quad (3.1)$$

$$\begin{aligned} \delta \mathbf{u}^s = & - \left(\mathbf{V}^{s\top} \frac{\partial^2 \mathcal{L}}{\partial \mathbf{u}^2} \mathbf{V}^s \right)^{-1} \mathbf{V}^{s\top} \left[\frac{\partial^2 \mathcal{L}}{\partial \boldsymbol{\theta} \partial \mathbf{u}} - \frac{\partial^2 \mathcal{L}}{\partial \mathbf{u}^2} \mathbf{V}^c \left(\frac{\partial \mathbf{G}}{\partial \mathbf{u}} \mathbf{V}^c \right)^{-1} \frac{\partial \mathbf{G}}{\partial \boldsymbol{\theta}} \right] \delta \boldsymbol{\theta} \\ & + o(\|\delta \boldsymbol{\theta}\|), \end{aligned} \quad (3.2)$$

where $\frac{\partial^2 \mathcal{L}}{\partial \mathbf{u}^2}$, $\frac{\partial^2 \mathcal{L}}{\partial \boldsymbol{\theta} \partial \mathbf{u}}$, $\frac{\partial \mathbf{G}}{\partial \mathbf{u}}$ and $\frac{\partial \mathbf{G}}{\partial \boldsymbol{\theta}}$ are calculated at the nominal solution point $\boldsymbol{\theta} = \boldsymbol{\theta}_\circ$.

Proof. By Theorem 2.3, there exist continuously differentiable vector functions $\mathbf{u}_\boldsymbol{\theta}^*$ and $\boldsymbol{\mu}_\boldsymbol{\theta}^*$ satisfying the second-order sufficient conditions for a local minimum of Problem (2.3) for each $\boldsymbol{\theta} \in \mathcal{B}_\eta(\boldsymbol{\theta}_\circ)$. The active constraints being linearly independent, and all the inequality constraints being assumed active with strictly positive Lagrange multipliers, we have

$$\mathbf{G}(\mathbf{u}_\boldsymbol{\theta}^*, \boldsymbol{\theta}) = \mathbf{0} \quad (3.3)$$

$$\frac{\partial \mathcal{L}}{\partial \mathbf{u}}(\mathbf{u}_\boldsymbol{\theta}^*, \boldsymbol{\mu}_\boldsymbol{\theta}^*, \boldsymbol{\theta}) = \frac{\partial \Phi}{\partial \mathbf{u}}(\mathbf{u}_\boldsymbol{\theta}^*, \boldsymbol{\theta}) + \boldsymbol{\mu}_\boldsymbol{\theta}^{*\top} \frac{\partial \mathbf{G}}{\partial \mathbf{u}}(\mathbf{u}_\boldsymbol{\theta}^*, \boldsymbol{\theta}) = \mathbf{0}, \quad (3.4)$$

for each $\boldsymbol{\theta} \in \mathcal{B}_\eta(\boldsymbol{\theta}_\circ)$.

The primal feasibility condition (3.3) gives

$$\begin{aligned} \mathbf{0} &= \mathbf{G}(\mathbf{u}_\boldsymbol{\theta}^*, \boldsymbol{\theta}) - \mathbf{G}(\mathbf{u}_{\boldsymbol{\theta}_\circ}^*, \boldsymbol{\theta}_\circ) \\ &= \frac{\partial \mathbf{G}}{\partial \boldsymbol{\theta}}(\mathbf{u}_{\boldsymbol{\theta}_\circ}^*, \boldsymbol{\theta}_\circ) \delta \boldsymbol{\theta} + \frac{\partial \mathbf{G}}{\partial \mathbf{u}}(\mathbf{u}_{\boldsymbol{\theta}_\circ}^*, \boldsymbol{\theta}_\circ) \delta \mathbf{u}^* + o(\|\delta \boldsymbol{\theta}\|) \\ &= \frac{\partial \mathbf{G}}{\partial \boldsymbol{\theta}}(\mathbf{u}_{\boldsymbol{\theta}_\circ}^*, \boldsymbol{\theta}_\circ) \delta \boldsymbol{\theta} + \frac{\partial \mathbf{G}}{\partial \mathbf{u}}(\mathbf{u}_{\boldsymbol{\theta}_\circ}^*, \boldsymbol{\theta}_\circ) \mathbf{V}^c \delta \mathbf{u}^c + \frac{\partial \mathbf{G}}{\partial \mathbf{u}}(\mathbf{u}_{\boldsymbol{\theta}_\circ}^*, \boldsymbol{\theta}_\circ) \mathbf{V}^s \delta \mathbf{u}^s \\ &\quad + o(\|\delta \boldsymbol{\theta}\|). \end{aligned}$$

Then, (3.1) follows by noting that $\frac{\partial \mathbf{G}}{\partial \mathbf{u}} \mathbf{V}^s = \mathbf{0}$ and $\frac{\partial \mathbf{G}}{\partial \mathbf{u}} \mathbf{V}^c = \mathbf{U} \Sigma^c$ is nonsingular at $\boldsymbol{\theta} = \boldsymbol{\theta}_\circ$, provided that $\mathbf{u}_{\boldsymbol{\theta}_\circ}^*$ is a regular point for the active constraints.

On the other hand, the dual feasibility condition (3.4), taken along the sensitivity-seeking directions, gives

$$\begin{aligned} \mathbf{0} &= \mathbf{V}^{s\top} \left[\frac{\partial \mathcal{L}}{\partial \mathbf{u}}(\mathbf{u}_\boldsymbol{\theta}^*, \boldsymbol{\mu}_\boldsymbol{\theta}^*, \boldsymbol{\theta}) - \frac{\partial \mathcal{L}}{\partial \mathbf{u}}(\mathbf{u}_{\boldsymbol{\theta}_\circ}^*, \boldsymbol{\mu}_{\boldsymbol{\theta}_\circ}^*, \boldsymbol{\theta}_\circ) \right]^\top \\ &= \mathbf{V}^{s\top} \left[\frac{\partial^2 \mathcal{L}}{\partial \boldsymbol{\theta} \partial \mathbf{u}}(\mathbf{u}_{\boldsymbol{\theta}_\circ}^*, \boldsymbol{\mu}_{\boldsymbol{\theta}_\circ}^*, \boldsymbol{\theta}_\circ) \delta \boldsymbol{\theta} + \frac{\partial^2 \mathcal{L}}{\partial \mathbf{u}^2}(\mathbf{u}_{\boldsymbol{\theta}_\circ}^*, \boldsymbol{\mu}_{\boldsymbol{\theta}_\circ}^*, \boldsymbol{\theta}_\circ) \delta \mathbf{u}^* \right. \\ &\quad \left. + \frac{\partial^2 \mathcal{L}}{\partial \boldsymbol{\mu} \partial \mathbf{u}}(\mathbf{u}_{\boldsymbol{\theta}_\circ}^*, \boldsymbol{\mu}_{\boldsymbol{\theta}_\circ}^*, \boldsymbol{\theta}_\circ) \delta \boldsymbol{\mu}^* \right] + o(\|\delta \boldsymbol{\theta}\|), \end{aligned}$$

where $\delta\boldsymbol{\mu}^* := \boldsymbol{\mu}_{\boldsymbol{\theta}}^* - \boldsymbol{\mu}_{\boldsymbol{\theta}_o}^*$. Since $\mathbf{V}^{s\top} \frac{\partial^2 \mathcal{L}}{\partial \boldsymbol{\mu} \partial \mathbf{u}} = \left[\frac{\partial \mathbf{G}}{\partial \mathbf{u}} \mathbf{V}^s \right]^\top = \mathbf{0}$, at $\boldsymbol{\theta} = \boldsymbol{\theta}_o$, we obtain²

$$\mathbf{0} = \mathbf{V}^{s\top} \left[\frac{\partial^2 \mathcal{L}}{\partial \boldsymbol{\theta} \partial \mathbf{u}} \delta \boldsymbol{\theta} + \frac{\partial^2 \mathcal{L}}{\partial \mathbf{u}^2} \mathbf{V}^c \delta \mathbf{u}^c + \frac{\partial^2 \mathcal{L}}{\partial \mathbf{u}^2} \mathbf{V}^s \delta \mathbf{u}^s \right] + o(\|\delta \boldsymbol{\theta}\|), \quad (3.5)$$

and from (3.1),

$$\begin{aligned} \mathbf{0} = \mathbf{V}^{s\top} \left[\frac{\partial^2 \mathcal{L}}{\partial \boldsymbol{\theta} \partial \mathbf{u}} - \frac{\partial^2 \mathcal{L}}{\partial \mathbf{u}^2} \mathbf{V}^c \left(\frac{\partial \mathbf{G}}{\partial \mathbf{u}} \mathbf{V}^c \right)^{-1} \frac{\partial \mathbf{G}}{\partial \boldsymbol{\theta}} \right] \delta \boldsymbol{\theta} + \left(\mathbf{V}^{s\top} \frac{\partial^2 \mathcal{L}}{\partial \mathbf{u}^2} \mathbf{V}^s \right) \delta \mathbf{u}^s \\ + o(\|\delta \boldsymbol{\theta}\|). \end{aligned}$$

Finally, (3.2) follows by noting that $\mathbf{V}^{s\top} \frac{\partial^2 \mathcal{L}}{\partial \mathbf{u}^2} \mathbf{V}^s$ is nonsingular at $\boldsymbol{\theta} = \boldsymbol{\theta}_o$, provided that the second-order sufficient conditions for a local minimum of Problem (2.3) for $\boldsymbol{\theta}_o$ hold at $\mathbf{u}_{\boldsymbol{\theta}_o}^*$. \square

In the presence of uncertainty, i.e., when the model parameters $\boldsymbol{\theta}$ deviate from their nominal values $\boldsymbol{\theta}_o$, failure to adapt the process inputs results in the cost value $\Phi(\mathbf{u}_{\boldsymbol{\theta}_o}^*, \boldsymbol{\theta})$. To combat uncertainty, adaptation of the process inputs can be made both in the constraint- and the sensitivity-seeking directions. The cost value corresponding to perfect input adaptation in the constraint-seeking directions is $\Phi(\mathbf{u}_{\boldsymbol{\theta}_o}^* + \mathbf{V}^c \delta \mathbf{u}^c, \boldsymbol{\theta})$, whereas perfect input adaptation in the sensitivity-seeking directions gives $\Phi(\mathbf{u}_{\boldsymbol{\theta}_o}^* + \mathbf{V}^s \delta \mathbf{u}^s, \boldsymbol{\theta})$. The cost variations $\delta \Phi^c$ and $\delta \Phi^s$ obtained upon adaptation of the process inputs in the constraint- and sensitivity-seeking directions, respectively, are thus given by

$$\begin{aligned} \delta \Phi_{\boldsymbol{\theta}}^c &:= \Phi(\mathbf{u}_{\boldsymbol{\theta}_o}^* + \mathbf{V}^c \delta \mathbf{u}^c, \boldsymbol{\theta}) - \Phi(\mathbf{u}_{\boldsymbol{\theta}_o}^*, \boldsymbol{\theta}), \\ \delta \Phi_{\boldsymbol{\theta}}^s &:= \Phi(\mathbf{u}_{\boldsymbol{\theta}_o}^* + \mathbf{V}^s \delta \mathbf{u}^s, \boldsymbol{\theta}) - \Phi(\mathbf{u}_{\boldsymbol{\theta}_o}^*, \boldsymbol{\theta}). \end{aligned}$$

Approximations of these variations are derived in the following theorem, based on the first-order approximations of the directional input variations established in Theorem 3.1.

Theorem 3.2 (Cost Variations in the Constraint-Seeking and Sensitivity-Seeking Directions)

Let the assumptions in Theorem 2.3 hold and η be chosen such that Theorem 2.3 applies. Then, the first- and second-order approximations of the cost variations $\delta \Phi_{\boldsymbol{\theta}}^c$ and $\delta \Phi_{\boldsymbol{\theta}}^s$, for $\boldsymbol{\theta} \in \mathcal{B}_\eta(\boldsymbol{\theta}_o)$, are given by

² For the sake of conciseness, the arguments $\mathbf{u}_{\boldsymbol{\theta}_o}^*, \boldsymbol{\mu}_{\boldsymbol{\theta}_o}^*, \boldsymbol{\theta}_o$ are dropped in the remainder of the proof.

$$\delta\Phi_{\theta}^c = \boldsymbol{\mu}_{\theta_{\circ}}^*{}^{\top} \frac{\partial \mathbf{G}}{\partial \boldsymbol{\theta}} \delta\boldsymbol{\theta} + o(\|\delta\boldsymbol{\theta}\|) \quad (3.6)$$

$$\delta\Phi_{\theta}^s =$$

$$\begin{aligned} & \delta\boldsymbol{\theta}^{\top} \left[\frac{1}{2} \frac{\partial^2 \mathcal{L}}{\partial \mathbf{u} \partial \boldsymbol{\theta}} \mathbf{V}^s (\mathbf{V}^{s\top} \frac{\partial^2 \mathcal{L}}{\partial \mathbf{u}^2} \mathbf{V}^s)^{-1} (\mathbf{V}^{s\top} \frac{\partial^2 \Phi}{\partial \mathbf{u}^2} \mathbf{V}^s) (\mathbf{V}^{s\top} \frac{\partial^2 \mathcal{L}}{\partial \mathbf{u}^2} \mathbf{V}^s)^{-1} \mathbf{V}^{s\top} \frac{\partial^2 \mathcal{L}}{\partial \boldsymbol{\theta} \partial \mathbf{u}} \right] \delta\boldsymbol{\theta} \\ & - \delta\boldsymbol{\theta}^{\top} \left[\frac{\partial^2 \Phi}{\partial \mathbf{u} \partial \boldsymbol{\theta}} \mathbf{V}^s (\mathbf{V}^{s\top} \frac{\partial^2 \mathcal{L}}{\partial \mathbf{u}^2} \mathbf{V}^s)^{-1} \mathbf{V}^{s\top} \frac{\partial^2 \mathcal{L}}{\partial \boldsymbol{\theta} \partial \mathbf{u}} \right] \delta\boldsymbol{\theta} + o(\|\delta\boldsymbol{\theta}\|^2), \end{aligned} \quad (3.7)$$

where $\frac{\partial \mathbf{G}}{\partial \boldsymbol{\theta}}$, $\frac{\partial^2 \Phi}{\partial \mathbf{u}^2}$, $\frac{\partial^2 \mathcal{L}}{\partial \mathbf{u}^2}$, $\frac{\partial^2 \Phi}{\partial \mathbf{u} \partial \boldsymbol{\theta}}$, and $\frac{\partial^2 \mathcal{L}}{\partial \mathbf{u} \partial \boldsymbol{\theta}}$ are calculated at the nominal solution point $\boldsymbol{\theta} = \boldsymbol{\theta}_{\circ}$.

Proof. Consider the variation in cost function $\delta\Phi_{\theta} := \Phi(\mathbf{u}_{\theta}^*, \boldsymbol{\theta}) - \Phi(\mathbf{u}_{\theta_{\circ}}^*, \boldsymbol{\theta}_{\circ})$. Its approximation up to second-order reads

$$\begin{aligned} \delta\Phi_{\theta} &= \frac{\partial \Phi}{\partial \mathbf{u}}(\mathbf{u}_{\theta_{\circ}}^*, \boldsymbol{\theta}_{\circ}) \delta\mathbf{u}^* + \delta\boldsymbol{\theta}^{\top} \frac{\partial^2 \Phi}{\partial \mathbf{u} \partial \boldsymbol{\theta}}(\mathbf{u}_{\theta_{\circ}}^*, \boldsymbol{\theta}_{\circ}) \delta\mathbf{u}^* \\ &+ \frac{1}{2} \delta\mathbf{u}^{*\top} \frac{\partial^2 \Phi}{\partial \mathbf{u}^2}(\mathbf{u}_{\theta_{\circ}}^*, \boldsymbol{\theta}_{\circ}) \delta\mathbf{u}^* + o(\|\delta\boldsymbol{\theta}\|^2), \end{aligned}$$

where $\delta\boldsymbol{\theta} := \boldsymbol{\theta} - \boldsymbol{\theta}_{\circ}$ and $\delta\mathbf{u}^* := \mathbf{u}_{\theta}^* - \mathbf{u}_{\theta_{\circ}}^*$.³ Splitting the input variations $\delta\mathbf{u}^*$ into constraint- and sensitivity-seeking directions, and noting that $\frac{\partial \Phi}{\partial \mathbf{u}} \mathbf{V}^s = \mathbf{0}$ at $\boldsymbol{\theta} = \boldsymbol{\theta}_{\circ}$, we obtain

$$\begin{aligned} \delta\Phi_{\theta} &= \frac{\partial \Phi}{\partial \mathbf{u}} \mathbf{V}^c \delta\mathbf{u}^c + \delta\boldsymbol{\theta}^{\top} \frac{\partial^2 \Phi}{\partial \mathbf{u} \partial \boldsymbol{\theta}} \mathbf{V}^c \delta\mathbf{u}^c + \delta\boldsymbol{\theta}^{\top} \frac{\partial^2 \Phi}{\partial \mathbf{u} \partial \boldsymbol{\theta}} \mathbf{V}^s \delta\mathbf{u}^s \quad (3.8) \\ &+ \frac{1}{2} \delta\mathbf{u}^{c\top} \mathbf{V}^{c\top} \frac{\partial^2 \Phi}{\partial \mathbf{u}^2} \mathbf{V}^c \delta\mathbf{u}^c + \frac{1}{2} \delta\mathbf{u}^{s\top} \mathbf{V}^{s\top} \frac{\partial^2 \Phi}{\partial \mathbf{u}^2} \mathbf{V}^s \delta\mathbf{u}^s \\ &+ \delta\mathbf{u}^{c\top} \mathbf{V}^{c\top} \frac{\partial^2 \Phi}{\partial \mathbf{u}^2} \mathbf{V}^s \delta\mathbf{u}^s + o(\|\delta\boldsymbol{\theta}\|^2). \end{aligned}$$

On the one hand, the cost variation $\delta\Phi_{\theta}^c$ is obtained by letting $\delta\mathbf{u}^s = \mathbf{0}$ in (3.8), and dropping the second-order terms,

$$\delta\Phi_{\theta}^c = \frac{\partial \Phi}{\partial \mathbf{u}} \mathbf{V}^c \delta\mathbf{u}^c + o(\|\delta\boldsymbol{\theta}\|)$$

From (3.1), we then get

$$\delta\Phi_{\theta}^c = - \left(\frac{\partial \Phi}{\partial \mathbf{u}} \mathbf{V}^c \right) \left(\frac{\partial \mathbf{G}}{\partial \mathbf{u}} \mathbf{V}^c \right)^{-1} \frac{\partial \mathbf{G}}{\partial \boldsymbol{\theta}} \delta\boldsymbol{\theta} + o(\|\delta\boldsymbol{\theta}\|).$$

and (3.6) follows by noting that $\boldsymbol{\mu}_{\theta_{\circ}}^*{}^{\top} = - \left(\frac{\partial \Phi}{\partial \mathbf{u}} \mathbf{V}^c \right) \left(\frac{\partial \mathbf{G}}{\partial \mathbf{u}} \mathbf{V}^c \right)^{-1}$ (dual feasibility condition along constraint-seeking directions).

On the other hand, the cost variation $\delta\Phi_{\theta}^s$ is obtained by letting $\delta\mathbf{u}^c = \mathbf{0}$ in (3.8),

³ For sake of conciseness, the arguments $\mathbf{u}_{\theta_{\circ}}^*$, $\boldsymbol{\theta}_{\circ}$ are dropped in the remainder of the proof.

$$\delta\Phi_{\theta}^s = \delta\theta^{\top} \frac{\partial^2 \Phi}{\partial \mathbf{u} \partial \theta} \mathbf{V}^s \delta \mathbf{u}^s + \frac{1}{2} \delta \mathbf{u}^{s\top} \mathbf{V}^{s\top} \frac{\partial^2 \Phi}{\partial \mathbf{u}^2} \mathbf{V}^s \delta \mathbf{u}^s + o(\|\delta\theta\|^2),$$

and (3.7) follows upon substitution of $\delta \mathbf{u}^s = -(\mathbf{V}^{s\top} \frac{\partial^2 \mathcal{L}}{\partial \mathbf{u}^2} \mathbf{V}^s)^{-1} \mathbf{V}^{s\top} \frac{\partial^2 \mathcal{L}}{\partial \theta \partial \mathbf{u}} \delta \theta + o(\|\delta\theta\|)$, as obtained from (3.5) with $\delta \mathbf{u}^c = \mathbf{0}$. \square

Overall, the variational analysis shows that failure to adapt the process inputs in the constraint-seeking directions results in cost variations in the order of the error $\delta\theta$; moreover, constraint violation can occur if the constraint-seeking directions are not adapted. On the other hand, failure to adapt the process inputs in the sensitivity-seeking directions gives cost variations proportional to the squared error $\delta\theta^2$ only. From a practical perspective, and provided that model mismatch remains moderate, more effort should therefore be placed on satisfying the process constraints rather than the sensitivity part of the NCO. Based on these considerations, an algorithm implementing constraints adaptation is presented in the following section.

3.2 Constraint Adaptation

3.2.1 Principles of Constraint Adaptation

The adaptation of additive constraint modifiers corresponds to the “classical” constraint-adaptation scheme [17, 33]. The constraints are corrected by simply offsetting the constraint predictions as

$$\mathbf{G}(\mathbf{u}, \theta) + \boldsymbol{\varepsilon} \leq \mathbf{0}, \quad (3.9)$$

where $\boldsymbol{\varepsilon} \in \mathbb{R}^{n_g}$ denotes the vector of the constraint correction factors or modifiers.

The decision variables are updated in each RTO iteration by solving an NLP problem similar to (2.3), which takes the constraint corrections into account:

$$\begin{aligned} \mathbf{u}_{k+1}^* &\in \arg \min_{\mathbf{u}} \Phi(\mathbf{u}, \theta) \\ \text{s.t.} \quad &\mathbf{G}(\mathbf{u}, \theta) + \boldsymbol{\varepsilon}_k \leq \mathbf{0} \\ &\mathbf{u}^L \leq \mathbf{u} \leq \mathbf{u}^U. \end{aligned} \quad (3.10)$$

Next, the optimal solution point \mathbf{u}_{k+1}^* is applied directly to the plant:

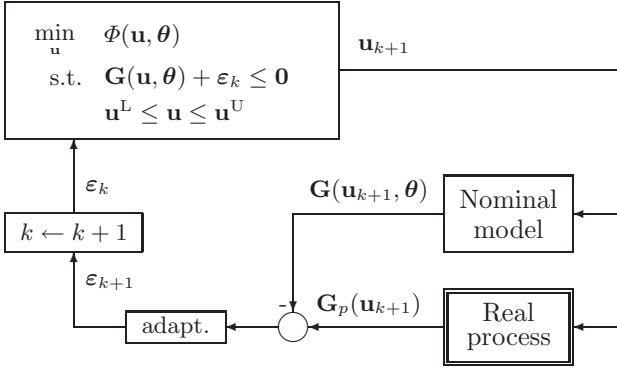


Fig. 3.1. Recursive adaptation of additive constraint modifiers.

$$\mathbf{u}_{k+1} = \mathbf{u}_{k+1}^*$$

Then, assuming that measurements are available for every constrained quantity at each RTO iteration, the constraint modifiers can be updated recursively as

$$\boldsymbol{\varepsilon}_{k+1} = \boldsymbol{\varepsilon}_k - \mathbf{B} [\mathbf{G}(\mathbf{u}_{k+1}, \boldsymbol{\theta}) + \boldsymbol{\varepsilon}_k - \mathbf{G}_p(\mathbf{u}_{k+1})], \quad (3.11)$$

where $\mathbf{B} \in \mathbf{R}^{n_g \times n_g}$ is a gain matrix. On the other hand, the model parameters $\boldsymbol{\theta}$ are kept constant at their nominal values.

Note that (3.11) can also be written as

$$\boldsymbol{\varepsilon}_{k+1} = (\mathbf{I} - \mathbf{B})\boldsymbol{\varepsilon}_k + \mathbf{B} (\mathbf{G}_p(\mathbf{u}_{k+1}) - \mathbf{G}(\mathbf{u}_{k+1}, \boldsymbol{\theta})), \quad (3.12)$$

which makes explicit the exponential filtering effect of \mathbf{B} . In the special case where \mathbf{B} is a diagonal matrix with entries b_i , $i = 1, \dots, n_g$, the update law can be seen as the filtered difference $G_{p,i}(\mathbf{u}_{k+1}) - G_i(\mathbf{u}_{k+1}, \boldsymbol{\theta})$ between the predicted and measured values of each constraint. No adaptation is performed for the i th constraint by setting $b_i = 0$, whereas no filtering is used for this constraint when $b_i = 1$, i.e., the full correction $G_{p,i}(\mathbf{u}_{k+1}) - G_i(\mathbf{u}_{k+1}, \boldsymbol{\theta})$ is applied.

The proposed constraint-adaptation algorithm is illustrated in Figure 3.1. Note that, unlike many other existing RTO schemes, the present approach relies on constraint measurements only, and it does not require that the gradient of the cost and constraint functions be estimated. In return, of course, the constraint-adaptation algorithm

may terminate at a suboptimal, yet feasible, point upon convergence. The extent of this loss of optimality is determined by the quality of the process model.

It may happen that some of the constraints are not measured during process operation, especially in those applications having very many constraints. In this case, observers that estimate the values of the unmeasured constrained quantities based on the available measurements (e.g., measured outputs \mathbf{y}_p) can be used. To avoid potential conflicts between the control and estimation tasks, however, great care must be taken to ensure that the rate of convergence of these observers is actually much faster than the RTO execution period itself. Furthermore, if the unmeasured constrained quantities cannot be estimated from plant measurements, one should opt for a more conservative approach, e.g., a robust optimization approach [96]. It is also possible to impose permanent (conservative) constraint backoffs to the unmeasured constraints [54].

The computational complexity of the NLP problems in constraint adaptation is similar to that of the classical two-step approach. Constraint adaptation is in fact less computationally demanding since it does not require the solution of a parameter estimation problem at each iteration.

3.2.2 Constraint-Adaptation Algorithm

We start the analysis with a number of definitions relative to the parametric programming problem (3.10) [6, 31]. The *feasible solution map*, $U(\cdot)$, is defined as

$$U(\boldsymbol{\varepsilon}) := \{\mathbf{u} \in [\mathbf{u}^L, \mathbf{u}^U] : \mathbf{G}(\mathbf{u}, \boldsymbol{\theta}) + \boldsymbol{\varepsilon} \leq \mathbf{0}\},$$

such that, for each $\boldsymbol{\varepsilon} \in \mathbb{R}^{n_g}$, $U(\boldsymbol{\varepsilon})$ is a subset of \mathbb{R}^{n_u} . A straightforward property of feasible solution sets is

$$U(\boldsymbol{\varepsilon}_1) \subset U(\boldsymbol{\varepsilon}_2), \quad \text{if } \boldsymbol{\varepsilon}_1 \geq \boldsymbol{\varepsilon}_2.$$

Observe also that $U(\boldsymbol{\varepsilon})$ may be empty for some values of $\boldsymbol{\varepsilon}$, which motivates the definition of the *domain* of $U(\cdot)$ as

$$\text{dom } U := \{\boldsymbol{\varepsilon} \in \mathbb{R}^{n_g} : U(\boldsymbol{\varepsilon}) \neq \emptyset\}.$$

Clearly, $\varepsilon_2 \in \text{dom } U$ for every $\varepsilon_2 \leq \varepsilon_1$ provided that $\varepsilon_1 \in \text{dom } U$.

In turn, the *optimal value function*, $\Phi^*(\cdot)$, and the *optimal solution map*, $U^*(\cdot)$, are defined as

$$\Phi^*(\varepsilon) := \begin{cases} \inf\{\Phi(\mathbf{u}, \boldsymbol{\theta}) : \mathbf{u} \in U(\varepsilon)\}, & \text{if } \varepsilon \in \text{dom } U \\ +\infty, & \text{otherwise} \end{cases}$$

$$U^*(\varepsilon) := \{\mathbf{u} \in U(\varepsilon) : \Phi(\mathbf{u}, \boldsymbol{\theta}) = \Phi^*(\varepsilon)\}.$$

The cost function $\Phi(\cdot, \boldsymbol{\theta})$ being continuous and the feasible region $U(\varepsilon)$ being bounded, it follows that the infimum $\Phi^*(\varepsilon)$ is assumed somewhere on $[\mathbf{u}^L, \mathbf{u}^U]$ for each $\varepsilon \in \text{dom } U$, thus being a minimum (Weierstrass' theorem – see, e.g., [7], Theorem 2.3.1). Besides the existence of a minimum point, we make the following uniqueness assumption throughout.

Assumption 3.1 *Let $\boldsymbol{\theta}$ be given. For each $\varepsilon \in \text{dom } U$, Problem (3.10) has a unique (global) solution point.*

This assumption implies that $U^*(\varepsilon)$ is a singleton, for each $\varepsilon \in \text{dom } U$. That is, at any RTO iteration k , we have $\mathbf{u}_{k+1}^* = U^*(\varepsilon_k)$, if $\varepsilon_k \in \text{dom } U$. Also, the map $\boldsymbol{\Gamma} : \text{dom } U \rightarrow \mathbb{R}^{n_g}$ representing the difference between the process constraints and the corrected values of the predicted constraints for a given correction ε ,

$$\boldsymbol{\Gamma}(\varepsilon) := \mathbf{G}(\mathbf{u}, \boldsymbol{\theta}) + \varepsilon - \mathbf{G}_p(\mathbf{u}), \quad \mathbf{u} = U^*(\varepsilon), \quad (3.13)$$

is a point-to-point map.

Remark 3.1 *In the more general case where Assumption 3.1 does not hold, $U^*(\varepsilon)$ may not be a singleton for $\varepsilon \in \text{dom } U$. That is, (4.15) defines a point-to-set map [45], $\boldsymbol{\Gamma} : \text{dom } U \rightarrow 2^{\mathbb{R}^{n_g}}$.*

Based on the foregoing definitions and Assumption 3.1, the (additive) constraint-adaptation algorithm \mathcal{M} can be stated as

$$\varepsilon_{k+1} = \mathcal{M}(\varepsilon_k), \quad (3.14)$$

where

$$\mathcal{M}(\varepsilon) := \varepsilon - \mathbf{B} \boldsymbol{\Gamma}(\varepsilon), \quad (3.15)$$

for any $\varepsilon \in \text{dom } U$.

3.2.3 Feasibility

An important property of the constraint-adaptation algorithm \mathcal{M} is that the iterates, upon convergence (in the absence of measurement noise and process disturbances), are guaranteed to terminate at a feasible point, under mild conditions. This is formalized in the following theorem:

Theorem 3.3 (Feasibility upon Convergence) *Let the gain matrix \mathbf{B} be nonsingular, and assume that the constraint-adaptation algorithm \mathcal{M} converges, with $\lim_{k \rightarrow \infty} \varepsilon_k = \varepsilon_\infty$. Then, $\mathbf{u}_\infty = U^*(\varepsilon_\infty)$ is a feasible operating point.*

Proof. Since \mathbf{B} is nonsingular, every fixed point ε_o of the algorithmic map $\mathcal{M}(\cdot)$ must satisfy

$$\Gamma(\varepsilon_o) = \mathbf{G}(\mathbf{u}_o, \boldsymbol{\theta}) + \varepsilon_o - \mathbf{G}_p(\mathbf{u}_o) = \mathbf{0},$$

where $\mathbf{u}_o = U^*(\varepsilon_o)$. In particular, ε_∞ being a fixed point of $\mathcal{M}(\cdot)$, we have

$$\mathbf{G}_p(\mathbf{u}_\infty) = \mathbf{G}(\mathbf{u}_\infty, \boldsymbol{\theta}) + \varepsilon_\infty \leq 0,$$

with $\mathbf{u}_\infty = U^*(\varepsilon_\infty)$. □

For certain problems, the iterates may converge by following an infeasible path (i.e., with violation of the constraints), even though the constraint-adaptation algorithm starts at a feasible point. A way of reducing the maximum violation of a given constraint, say $G_{p,i}$, is by decreasing the corresponding filter parameter b_i (in the case where \mathbf{B} is a diagonal matrix). Yet, this is at the expense of a slower convergence rate.

Even though the iterates may follow an infeasible path, a straightforward corollary of Theorem 3.3 is that the constraint-adaptation algorithm, provided it converges, can yield feasible operation after a *finite* number of RTO periods upon backing-off the constraints of the original RTO problem.

3.2.4 Active Set

Owing to the importance of constraints in optimization problems, a very desirable property of any RTO scheme is the capability of detecting the optimal set of active constraints. Since the constraint-adaptation algorithm approximates the actual gradients of the cost

Table 3.1. Values of the uncertain parameters $\boldsymbol{\theta}$ in Problem (3.16) corresponding to the model and two simulated realities for the plant.

	Model	Plant A	Plant B
θ_1	0.75	0.75	0.75
θ_2	1.40	2.00	2.00
θ_3	1.00	1.00	1.80

and constraint functions by using a fixed process model, it cannot be guaranteed that a point satisfying the complementarity slackness and dual feasibility conditions (2.6–2.8) is attained. Observe, however, that the active set determined by the process model is known to be the same as the actual optimal active set, provided that the model mismatch remains moderate (see Theorem 2.3). There is therefore much hope that the constraint-adaptation algorithm can provide a good approximation of the correct active set, even in the case of large model mismatch. These important considerations are illustrated next through a numerical example; they are further illustrated by the case study application presented in Subsection 3.2.8.

Example 3.1 Consider the following quadratic program (QP):

$$\begin{aligned} \min \quad & \Phi(\mathbf{u}, \boldsymbol{\theta}) := (u_1 - \theta_1)^2 + (u_2 - \theta_1)^2 & (3.16) \\ \text{s.t.} \quad & G_1 := u_1 - \theta_2(1 - u_2) \leq 0 \\ & G_2 := u_2\theta_3 - 2(1 - u_1) \leq 0, \end{aligned}$$

with two decision variables $\mathbf{u} = [u_1 \ u_2]^\top$, three model parameters $\boldsymbol{\theta} = [\theta_1 \ \theta_2 \ \theta_3]^\top$, and two inequality constraints. The parameter values $\boldsymbol{\theta}$ for the model and the simulated realities A and B for the plant are reported in Table 3.1.

We start by considering the plant A. Since the parameters θ_1 and θ_3 are identical in the model and the plant, no adaptation is needed for G_2 (i.e., $\varepsilon_2 = 0$); however, this information is not known by the constraint-adaptation algorithm. Furthermore, the cost functions of the model and the plant are identical, i.e., $\Phi(\mathbf{u}, \boldsymbol{\theta}) \equiv \Phi_p(\mathbf{u})$. Hence, the only uncertainty is in the constraint G_1 . Upon application of the constraint-adaptation algorithm \mathcal{M} with $\mathbf{B} = \mathbf{I}_{2 \times 2}$ and $\boldsymbol{\varepsilon}_0 = [0 \ 0]^\top$, the iterates converge to the fixed point $\boldsymbol{\varepsilon}_\infty = [-\frac{1}{5} \ 0]^\top$. The correspond-

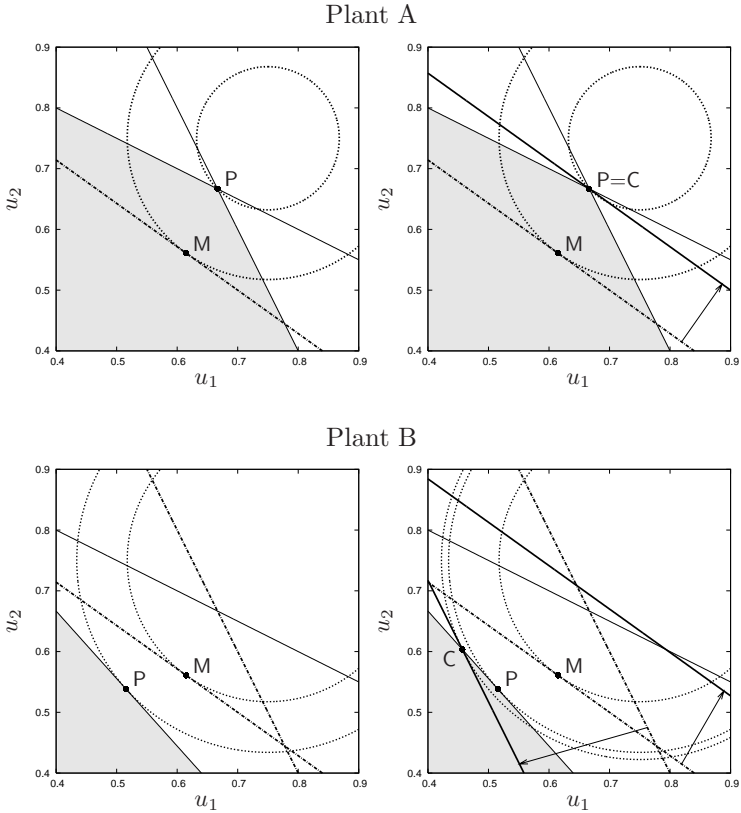


Fig. 3.2. Illustration of the constraint-adaptation algorithm for Problem (3.16). **Left plots:** no constraint adaptation; **Right plots:** converged constraint adaptation; Thin solid lines: constraint bounds for the plant; Colored area: feasible region; Thick dash-dotted lines: constraint bounds predicted by the model without adaptation; Thick solid lines: constraint bounds predicted by the model upon convergence of the constraint-adaptation algorithm; Dotted lines: contours of the cost function; Point P: optimum for the plant; Point M: optimum for the model without adaptation; Point C: optimum for model upon convergence of the constraint-adaptation algorithm; Arrows: direction of constraint adaptation.

ing operating point \mathbf{u}_∞ lies at the intersection of the constraints $G_{p,1}$ and $G_{p,2}$ (see Figure 3.2, Plant A). Note that the correct optimal point would also be found in the case where $\theta_1 \neq \underline{\theta}_1$, as long as both inequality

constraints remain active at this point. In the plant B, both the constraints G_1 and G_2 are uncertain. Observe that adaptation is absolutely warranted in this scenario, since the solution of the model-based optimization problem (3.16) leads to infeasible operation. The constraint-adaptation algorithm converges to the fixed point $\varepsilon_\infty = [-\frac{69}{290} \frac{14}{29}]^\top$, with the same initial guess and gain matrix as previously. The corresponding operating point lies on the constraint G_2 at the point of tangency with the cost contours, while the constraint G_1 is inactive. Here again, the correct active set is detected upon convergence of the constraint-adaptation algorithm, yet the iterates converge to a suboptimal point (see Figure 3.2, Plant B). The loss of performance, around 7%, is due to the error made in evaluating the gradient of the active constraint G_2 . This loss is rather limited in view of the substantial variation in the model parameters θ_2 and θ_3 .

3.2.5 Convergence

A particularly important aspect of any RTO scheme relates to its convergence properties. It has been shown earlier that, upon convergence, the constraint-adaptation algorithm \mathcal{M} terminates at a feasible operating point, under mild conditions. Yet, this algorithm may not systematically converge. For example, it may happen that the NLP problem (3.10) becomes infeasible because a coefficient of the constraint modifiers ε has grown too big. It can also happen that a coefficient of the constraint modifiers oscillates between two values because the corresponding filter parameter is too large.

In the absence of measurement noise and process disturbances, the following theorem provides a necessary condition for the convergence of \mathcal{M} to a fixed point, and it establishes the convergence rate.

Theorem 3.4 (Necessary Condition and Convergence Rate)

Let ε_∞ be a fixed point of the constraint-adaptation algorithm \mathcal{M} , and assume that:

1. the second-order sufficient conditions for a local minimum of Problem (3.10) with $\varepsilon = \varepsilon_\infty$ hold at $\mathbf{u}_\infty = U^*(\varepsilon_\infty)$, with the associated Lagrange multipliers $\boldsymbol{\mu}_\infty$, ζ_∞^U , and ζ_∞^L ;
2. \mathbf{u}_∞ is a regular point for the active constraints, i.e. LICQ holds;
3. $\mu_{i,\infty} > 0$, $\zeta_{i,\infty}^U > 0$, and $\zeta_{i,\infty}^L > 0$ for each active constraint, i.e. strict complementary slackness holds.

Then, a necessary condition for the local convergence of \mathcal{M} to the fixed-point ε_∞ is that the gain matrix \mathbf{B} be chosen such that

$$\varrho \left\{ \mathbf{I} - \mathbf{B} \left[\mathbf{I} + \left(\frac{\partial \mathbf{G}_p}{\partial \mathbf{u}}(\mathbf{u}_\infty) - \frac{\partial \mathbf{G}}{\partial \mathbf{u}}(\mathbf{u}_\infty, \boldsymbol{\theta}) \right) \mathcal{P}_\mathbf{u} \mathbf{M}_\infty^{-1} \mathbf{N}_\infty \right] \right\} < 1, \quad (3.17)$$

where the matrices $\mathbf{M}_\infty \in \mathbb{R}^{(n_g+3n_u) \times (n_g+3n_u)}$, $\mathbf{N}_\infty \in \mathbb{R}^{(n_g+3n_u) \times n_g}$, and $\mathcal{P}_\mathbf{u} \in \mathbb{R}^{n_u \times (n_g+3n_u)}$ are defined as

$$\mathbf{M}_\infty := \begin{pmatrix} \frac{\partial^2 \mathcal{L}}{\partial \mathbf{u} \partial \mathbf{u}} & \frac{\partial G_{p,1}}{\partial \mathbf{u}}^\top & \dots & \frac{\partial G_{p,n_g}}{\partial \mathbf{u}}^\top & \mathbf{I}_{n_u \times n_u} & -\mathbf{I}_{n_u \times n_u} \\ \mu_{1,\infty} \frac{\partial G_{p,1}}{\partial \mathbf{u}} & G_{p,1} & & & & \\ \vdots & & \ddots & & & \\ \mu_{n_g,\infty} \frac{\partial G_{p,n_g}}{\partial \mathbf{u}} & & & G_{p,n_g} & & \\ \text{diag}(\boldsymbol{\zeta}_\infty^U) & & & & \text{diag}(\mathbf{u} - \mathbf{u}^U) & \\ -\text{diag}(\boldsymbol{\zeta}_\infty^L) & & & & & \text{diag}(\mathbf{u}^L - \mathbf{u}) \end{pmatrix}_{(\mathbf{u}_\infty)}$$

$$\mathbf{N}_\infty := \begin{pmatrix} \mathbf{0}_{n_g \times n_u} & -\text{diag}(\boldsymbol{\mu}_\infty) & \mathbf{0}_{n_g \times 2n_u} \end{pmatrix}^\top,$$

$$\mathcal{P}_\mathbf{u} := \begin{pmatrix} \mathbf{I}_{n_u \times n_u} & \mathbf{0}_{n_u \times (n_g+2n_u)} \end{pmatrix},$$

and $\varrho\{\cdot\}$ stands for the spectral radius. Moreover, if the constraint adaptation converges, then the rate of convergence is linear.

Proof. It follows from the assumptions and Theorem 2.3 that there is some $\eta > 0$ such that, for each $\varepsilon \in \mathcal{B}_\eta(\varepsilon_\infty)$, there exists a unique continuously differentiable vector function $\mathbf{u}^* = U^*(\varepsilon)$ satisfying the second-order sufficient conditions for a local minimum of Problem (3.10), with $\mathbf{u}_\infty = U^*(\varepsilon_\infty)$. Moreover, we have (see Theorem 2.3):

$$\frac{\partial U^*}{\partial \varepsilon}(\varepsilon_\infty) = -\mathcal{P}_\mathbf{u} \mathbf{M}_\infty^{-1} \mathbf{N}_\infty.$$

The constraint functions $G_i(\cdot, \boldsymbol{\theta})$ and $G_{p,i}(\cdot)$, $i = 1, \dots, n_g$, being differentiable with respect to \mathbf{u} , it follows that $\boldsymbol{\Gamma}(\cdot)$ given by (3.13) is itself differentiable with respect to ε in a neighborhood of ε_∞ , and we have

$$\begin{aligned} \frac{\partial \boldsymbol{\Gamma}}{\partial \varepsilon}(\varepsilon_\infty) &= \mathbf{I} - \left(\frac{\partial \mathbf{G}_p}{\partial \mathbf{u}}(\mathbf{u}_\infty) - \frac{\partial \mathbf{G}}{\partial \mathbf{u}}(\mathbf{u}_\infty, \boldsymbol{\theta}) \right) \frac{\partial U^*}{\partial \varepsilon}(\varepsilon_\infty) \\ &= \mathbf{I} + \left(\frac{\partial \mathbf{G}_p}{\partial \mathbf{u}}(\mathbf{u}_\infty) - \frac{\partial \mathbf{G}}{\partial \mathbf{u}}(\mathbf{u}_\infty, \boldsymbol{\theta}) \right) \mathcal{P}_\mathbf{u} \mathbf{M}_\infty^{-1} \mathbf{N}_\infty. \end{aligned}$$

That is, a first-order approximation of Γ in a neighborhood of ε_∞ is

$$\Gamma(\varepsilon) = \left[\mathbf{I} + \left(\frac{\partial \mathbf{G}_p}{\partial \mathbf{u}}(\mathbf{u}_\infty) - \frac{\partial \mathbf{G}}{\partial \mathbf{u}}(\mathbf{u}_\infty, \boldsymbol{\theta}) \right) \mathcal{P}_\mathbf{u} \mathbf{M}_\infty^{-1} \mathbf{N}_\infty \right] \delta \varepsilon + o(\|\delta \varepsilon\|),$$

where $\delta \varepsilon := \varepsilon - \varepsilon_\infty$.

Next, suppose that the algorithm (3.14), with ε_0 given, converges to ε_∞ . Then,

$$\exists k_0 > 0 \text{ such that } \varepsilon_k \in \mathcal{B}_\eta(\varepsilon_\infty), \forall k > k_0,$$

and we have

$$\delta \varepsilon_{k+1} = \Upsilon_\infty^{\mathbf{G}} \delta \varepsilon_k + o(\|\delta \varepsilon_k\|), \quad (3.18)$$

for each $k > k_0$, with

$$\Upsilon_\infty^{\mathbf{G}} := \mathbf{I} - \mathbf{B} \left[\mathbf{I} + \left(\frac{\partial \mathbf{G}_p}{\partial \mathbf{u}}(\mathbf{u}_\infty) - \frac{\partial \mathbf{G}}{\partial \mathbf{u}}(\mathbf{u}_\infty, \boldsymbol{\theta}) \right) \mathcal{P}_\mathbf{u} \mathbf{M}_\infty^{-1} \mathbf{N}_\infty \right]. \quad (3.19)$$

For \mathcal{M} to converge to ε_∞ , the spectral radius of $\Upsilon_\infty^{\mathbf{G}}$ must therefore be less than 1. Finally, it follows from (3.18) that, if \mathcal{M} converges, then it has a linear rate of convergence. \square

Remark 3.2 *It follows from the local analysis performed in the proof of Theorem 3.4 that a necessary condition for the iterates ε_k to converge to the fixed point ε_∞ monotonically, i.e. without oscillations around ε_∞ , is that all the eigenvalues of matrix $\Upsilon_\infty^{\mathbf{G}}$ be nonnegative.*

We illustrate these results in the following example.

Example 3.2 *Consider the QP problem (3.16) of Example 3.1, with the parameters specified in Table 3.1. For the plant A , the converged constraint modifiers as well as the corresponding inputs and multipliers are:*

$$\varepsilon_\infty = \begin{pmatrix} -\frac{1}{5} \\ 0 \end{pmatrix}, \quad \mathbf{u}^*(\varepsilon_\infty) = \begin{pmatrix} \frac{2}{3} \\ \frac{3}{3} \\ \frac{1}{3} \end{pmatrix}, \quad \boldsymbol{\mu}^*(\varepsilon_\infty) = \begin{pmatrix} \frac{5}{1} \\ \frac{54}{27} \end{pmatrix}.$$

In particular, it can be verified that the second-order sufficient conditions hold at $\mathbf{u}^(\varepsilon_\infty)$. The derivatives $\frac{\partial \mathbf{u}^*}{\partial \varepsilon}$ and $\frac{\partial \Gamma}{\partial \varepsilon}$ at ε_∞ are obtained as*

$$\frac{\partial \mathbf{u}^*}{\partial \varepsilon}(\varepsilon_\infty) = \frac{1}{9} \begin{pmatrix} -\frac{5}{9} & \frac{7}{9} \\ \frac{10}{9} & -\frac{5}{9} \end{pmatrix}, \quad \frac{\partial \Gamma}{\partial \varepsilon}(\varepsilon_\infty) = \begin{pmatrix} \frac{5}{3} & -\frac{1}{3} \\ 0 & 1 \end{pmatrix}.$$

Upon choosing the diagonal gain matrix as $\mathbf{B} := b\mathbf{I}$, the matrix $\mathbf{r}_\infty^{\mathbf{G}}$ reads

$$\mathbf{r}_\infty^{\mathbf{G}} = \begin{pmatrix} 1 - \frac{5}{3}b & \frac{1}{3}b \\ 0 & 1 - b \end{pmatrix}.$$

Clearly, the necessary condition (4.55) is satisfied for every $0 < b < \frac{6}{5}$. Note also that monotonic convergence is ensured as long as $0 < b \leq \frac{3}{5}$ (see Remark 3.2).

The foregoing analysis provides conditions that are necessary for the constraint-adaptation algorithm to converge. Yet, these conditions are inherently local and, in general, are not sufficient for convergence. This is the case, e.g., when the algorithm starts far away from a fixed point. Another source of divergence for the constraint-adaptation algorithm is when the constraint map $\mathbf{\Gamma}(\varepsilon)$ is always nonzero, i.e., the algorithm does not have a fixed point. One such example is discussed subsequently.

Table 3.2. Values of the uncertain parameters $\boldsymbol{\theta}$ in Problem (3.20) corresponding to the model and the plant.

	Model Plant	
θ_1	0.5	0.5
θ_2	1.1	0.2
θ_3	1.0	-1.0

Example 3.3 Consider the convex QP:

$$\min_{\mathbf{u} \geq \mathbf{0}} \Phi(\mathbf{u}, \boldsymbol{\theta}) = (u_1 - \theta_1)^2 + (u_2 - \theta_1)^2 \quad (3.20)$$

$$s.t. \quad G := \theta_2 - u_1 - u_2\theta_3 \leq 0$$

with two decision variables $\mathbf{u} = [u_1, u_2]$, three uncertain parameters $\boldsymbol{\theta} = [\theta_1, \theta_2, \theta_3]$ whose values corresponding to the model and the plant are given in Table 3.2, and a single constraint to be adapted using the constraint modifier ε .

Upon application of the constraint-adaptation algorithm with a scalar gain $b > 0$, the iterates are found to diverge, $\varepsilon_k \rightarrow +\infty$ as

$k \rightarrow +\infty$. The situation is illustrated in the left plot of Figure 3.3. Here, divergence results from the fact that the optimal solution set $U^*(\varepsilon)$ does not intersect the plant feasible region, irrespective of the constraint modifier ε . That is, the constraint map $\Gamma(\varepsilon)$ shown in the right plot of Figure 3.3 is always nonzero. Clearly, the necessary conditions given in Theorem 3.4 do not hold for this problem.

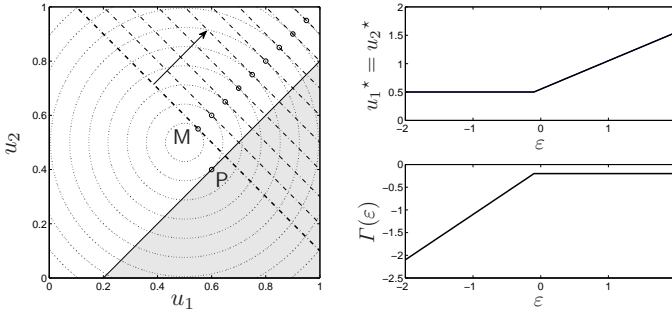


Fig. 3.3. Divergence of the constraint-adaptation algorithm for Problem (3.20). **Left plot:** iterates in the u_1 - u_2 space; **Thin solid lines:** constraint bounds for the plant; **Colored Area:** feasible region; **Thick dash-dotted lines:** constraint bounds predicted by the model without adaptation; **Thin dash-dotted lines:** constraint bounds upon application of the constraint-adaptation algorithm; **Dotted lines:** contours of the cost function; **Point P:** optimum for the plant; **Point M:** optimum for the model without adaptation; **Arrow:** direction of constraint adaptation; **Right plots:** optimal solution point $\mathbf{u}^*(\varepsilon)$ and constraint map $\Gamma(\varepsilon)$ versus ε .

3.2.6 Effect of Measurement Noise

To analyze the effect of measurement errors on the convergence of the constraint-adaptation algorithm, suppose that the constrained quantities during the k th RTO period are measured as $\mathbf{G}_p(\mathbf{u}_k^*) + \boldsymbol{\nu}_k$ instead of $\mathbf{G}_p(\mathbf{u}_k^*)$. From (3.11), the measurement errors $\boldsymbol{\nu}_k$ give rise to the following errors in the constraint modifiers:

$$\Delta \boldsymbol{\varepsilon}_{k+1} = -\mathbf{B}\boldsymbol{\nu}_k.$$

In turn, provided that the constraint modifiers $\boldsymbol{\varepsilon}_{k+1}$ are in a neighborhood of the fixed point $\boldsymbol{\varepsilon}_\infty$, a first-order approximation of the input deviations is obtained as

$$\Delta \mathbf{u}_{k+1}^* = -\frac{\partial U^*}{\partial \boldsymbol{\varepsilon}}(\boldsymbol{\varepsilon}_\infty) \mathbf{B} \boldsymbol{\nu}_k = \mathcal{P}_{\mathbf{u}} \mathbf{M}_\infty^{-1} \mathbf{N}_\infty \mathbf{B} \boldsymbol{\nu}_k + o(\|\mathbf{B} \boldsymbol{\nu}_k\|).$$

Accordingly, variations in both the constraint modifiers and the process inputs induced by measurement noise can be reduced by tuning the gain matrix \mathbf{B} . In the case of a diagonal gain matrix, this is done most easily by decreasing the gain coefficients b_1, \dots, b_{n_g} . Yet, this is at the expense of a slower convergence, and a tradeoff must therefore be found between the level of filtering and the speed of convergence of the algorithm. In particular, observe that $\Delta \mathbf{u}_{k+1}^*$ may still be very large although $\Delta \boldsymbol{\varepsilon}_{k+1}$ is small, e.g., when some of the sensitivity coefficients $\frac{\partial U^*}{\partial \boldsymbol{\varepsilon}}(\boldsymbol{\varepsilon}_\infty)$ of the optimal inputs to the constraint modifiers are large.

3.2.7 Alternative Constraint-Adaptation Algorithm

In this subsection, an alternative way of adapting the constraints is introduced where the constraint bounds are directly updated based on the measured values of the constrained variables. The resulting filtering of the constraint modifiers $\boldsymbol{\varepsilon}$ is no longer recursive as in (3.12). If at each RTO iteration k the optimal inputs evaluated by the RTO optimizer are applied directly to the plant, the alternative constraint adaptation presented in this subsection does not present any particular advantage with respect to the adaptation given by (3.9, 3.12). However, the alternative adaptation is introduced here, as it is preferred when the optimal inputs given by the RTO level are implemented through a constraint controller, as in Section 3.3.

In the homogeneous form, the process constraints are expressed as $\mathbf{G}(\mathbf{u}, \boldsymbol{\theta}) \leq \mathbf{0}$, with zeros at the right side of the inequality. In this subsection, since the constraint bounds are directly adapted, it is preferred to work with the process constraints expressed as $\bar{\mathbf{G}}(\mathbf{u}, \boldsymbol{\theta}) \leq \bar{\mathbf{G}}^\mathbf{U}$, where $\bar{\mathbf{G}} \in \mathbb{R}^{n_g}$ are the constrained quantities, and $\bar{\mathbf{G}}^\mathbf{U} \in \mathbb{R}^{n_g}$ are the constraint bounds. With this notation, $\mathbf{G}(\mathbf{u}, \boldsymbol{\theta}) = \bar{\mathbf{G}}(\mathbf{u}, \boldsymbol{\theta}) - \bar{\mathbf{G}}^\mathbf{U}$. Constraint adaptation (3.9, 3.12) represents the *classical constraint-adaptation scheme*, which is rewritten below in terms of the constrained quantities $\bar{\mathbf{G}}$:

$$\begin{aligned}
\mathbf{u}_{k+1}^* &\in \arg \min_{\mathbf{u}} \Phi(\mathbf{u}, \boldsymbol{\theta}) & (3.21) \\
\text{s.t. } & \bar{\mathbf{G}}(\mathbf{u}, \boldsymbol{\theta}) + \boldsymbol{\varepsilon}_k \leq \bar{\mathbf{G}}^U \\
& \mathbf{u}^L \leq \mathbf{u} \leq \mathbf{u}^U
\end{aligned}$$

The optimal solution point \mathbf{u}_{k+1}^* is applied directly to the plant, i.e., $\mathbf{u}_{k+1} = \mathbf{u}_{k+1}^*$. Subsequently, the constraint modifiers are updated recursively as in (3.12).

Next, a different way of adapting the constraints in the optimization problem is introduced:

$$\bar{\mathbf{G}}(\mathbf{u}, \boldsymbol{\theta}) + \boldsymbol{\gamma}_k^G \leq \bar{\mathbf{G}}_k^U, \quad (3.22)$$

where the correction term

$$\boldsymbol{\gamma}_k^G := \bar{\mathbf{G}}_p(\mathbf{u}_k) - \bar{\mathbf{G}}(\mathbf{u}_k, \boldsymbol{\theta}), \quad (3.23)$$

stands for the difference between the measured and predicted values of the constraints. At the current RTO iteration, the constraint bounds $\bar{\mathbf{G}}_k^U$ are updated directly from the measured values of the constraints as:

$$\bar{\mathbf{G}}_k^U = \bar{\mathbf{G}}_p(\mathbf{u}_k) + \mathbf{B} \left(\bar{\mathbf{G}}^U - \bar{\mathbf{G}}_p(\mathbf{u}_k) \right), \quad (3.24)$$

where $\mathbf{B} \in \mathbb{R}^{n_g \times n_g}$ is a diagonal gain matrix with entries $b_i \in (0, 1]$, $i = 1, \dots, n_g$. In this approach, the modified optimization problem reads:

$$\begin{aligned}
\mathbf{u}_{k+1}^* &\in \arg \min_{\mathbf{u}} \Phi(\mathbf{u}, \boldsymbol{\theta}) & (3.25) \\
\text{s.t. } & \bar{\mathbf{G}}(\mathbf{u}, \boldsymbol{\theta}) + \boldsymbol{\gamma}_k^G \leq \bar{\mathbf{G}}_k^U \\
& \mathbf{u}^L \leq \mathbf{u} \leq \mathbf{u}^U
\end{aligned}$$

The optimal solution point \mathbf{u}_{k+1}^* is applied directly to the plant, i.e., $\mathbf{u}_{k+1} = \mathbf{u}_{k+1}^*$. Note that, in this alternative constraint-adaptation scheme, the constraint modifiers $\boldsymbol{\varepsilon}$ are not updated recursively as in (3.12). Instead, they are obtained as follows:

$$\boldsymbol{\varepsilon}_k = \bar{\mathbf{G}}^U - \bar{\mathbf{G}}_k^U + \boldsymbol{\gamma}_k^G, \quad (3.26)$$

with $\boldsymbol{\gamma}_k^G$ given by (3.23) and $\bar{\mathbf{G}}_k^U$ given by (3.24). For the combination with constraint control, constraint adaptation (3.22-3.24) is preferred

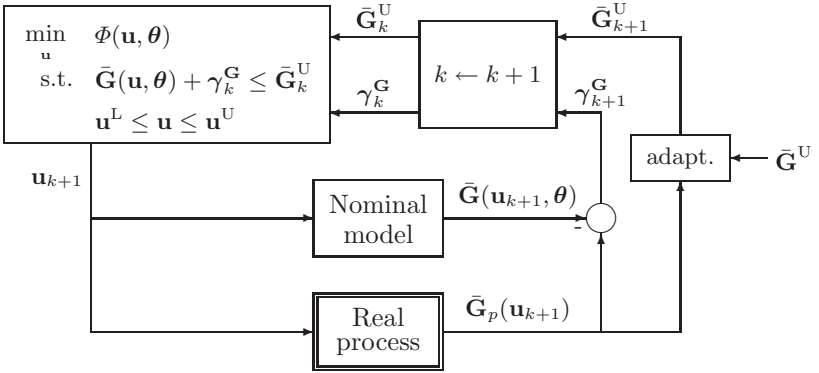


Fig. 3.4. Alternative constraint-adaptation scheme.

because it gives the ability to vary the setpoints $\bar{\mathbf{G}}_k^U$ passed to the constraint controller. The proposed alternative constraint-adaptation scheme is illustrated in Figure 3.4.

With the constraint modifiers defined as in (3.26), the feasible solution map $U(\cdot)$ and its domain, as well as the optimal value function $\Phi^*(\cdot)$, and the optimal solution map $U^*(\cdot)$ can be defined in a similar way as in Subsection 3.2.2. Furthermore, by Assumption 3.1, for each $\varepsilon \in \text{dom } U$, Problem (3.25) has a unique (global) solution point, $\mathbf{u}^* = U^*(\varepsilon)$.

Feasibility

If the iterates of the alternative constraint-adaptation algorithm represented in Figure 3.4 converge (in the absence of measurement noise and process disturbances), they are guaranteed, under mild conditions, to terminate at a feasible point. This is formalized in the following theorem:

Theorem 3.5 (Feasibility upon Convergence) *Let the gain matrix \mathbf{B} be nonsingular and the alternative constraint-adaptation algorithm given by (3.23), (3.24) and (3.25) (see the algorithm scheme in Figure 3.4) converge, with $\lim_{k \rightarrow \infty} \varepsilon_k = \varepsilon_\infty$. Then, $\mathbf{u}_\infty = U^*(\varepsilon_\infty)$ is a feasible operating point.*

Proof. Taking the limit when $k \rightarrow \infty$ of (3.26) we have:

$$\begin{aligned}\varepsilon_\infty &= \bar{\mathbf{G}}^U - \bar{\mathbf{G}}_\infty^U + \gamma_\infty^{\mathbf{G}} \\ &= \bar{\mathbf{G}}^U - \mathbf{B} \left(\bar{\mathbf{G}}^U - \bar{\mathbf{G}}_p(\mathbf{u}_\infty) \right) - \bar{\mathbf{G}}(\mathbf{u}_\infty, \boldsymbol{\theta}).\end{aligned}\quad (3.27)$$

But $\bar{\mathbf{G}}(\mathbf{u}_\infty, \boldsymbol{\theta}) + \varepsilon_\infty \leq \bar{\mathbf{G}}^U$ since $\mathbf{u}_\infty = U^*(\varepsilon_\infty)$, and therefore

$$\mathbf{B} \left(\bar{\mathbf{G}}^U - \bar{\mathbf{G}}_p(\mathbf{u}_\infty) \right) \geq \mathbf{0}$$

Finally, \mathbf{B} being nonsingular, we obtain

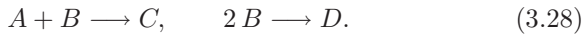
$$\bar{\mathbf{G}}_p(\mathbf{u}_\infty) \leq \bar{\mathbf{G}}^U,$$

thereby showing that \mathbf{u}_∞ is a feasible operating point. \square

However, the constraints can be violated during the iterations, even when the alternative constraint-adaptation algorithm starts at a feasible point. Infeasibility during the iterations will be addressed in Subsections 3.3.1 and 3.3.2 upon combination of the alternative constraint-adaptation scheme with a constraint controller.

3.2.8 Case Study: Isothermal CSTR

The example presented in [79] is considered to illustrate the constraint-adaptation algorithm. It consists of an isothermal continuous stirred-tank reactor with two reactions:



The desired product is C , while D is undesired. The reactor is fed by two streams with the flow rates F_A and F_B and the corresponding inlet concentrations $c_{A_{in}}$ and $c_{B_{in}}$.

Model Equations and Parameters

The steady-state model results from material balance equations:

$$F_A c_{A_{in}} - (F_A + F_B) c_A - r_1 V = 0 \quad (3.29)$$

$$F_B c_{B_{in}} - (F_A + F_B) c_B - r_1 V - 2r_2 V = 0 \quad (3.30)$$

$$-(F_A + F_B) c_C + r_1 V = 0, \quad (3.31)$$

with

$$r_1 = k_1 c_A c_B, \quad r_2 = k_2 c_B^2. \quad (3.32)$$

The heat produced by the chemical reactions is:

$$q_r = (-\Delta H_1)r_1V + (-\Delta H_2)r_2V. \quad (3.33)$$

Variables and parameters: c_X : concentration of species X , V : volume, r_i : rate of reaction i , k_i : kinetic coefficient of reaction i , ΔH_i : enthalpy of reaction i .

Table 3.3. Nominal model parameters and parameter bounds

k_1	1.5	$\frac{1}{\text{mol}\cdot\text{h}}$	k_2	0.014	$\frac{1}{\text{mol}\cdot\text{h}}$
$c_{A_{in}}$	2	$\frac{\text{mol}}{1}$	$c_{B_{in}}$	1.5	$\frac{\text{mol}}{1}$
ΔH_1	-7×10^4	$\frac{\text{J}}{\text{mol}}$	ΔH_2	-10^5	$\frac{\text{J}}{\text{mol}}$
V	500	1	$q_{r,\max}$	10^6	$\frac{\text{J}}{\text{h}}$
F_{\max}	22	$\frac{1}{\text{h}}$			

The numerical values of the parameters are given in Table 3.3. In the sequel, k_1 denotes the value of the kinetic parameter used in the process model,⁴ whereas \bar{k}_1 is the plant (simulated reality) value.

Optimization Problem

The cost function is chosen as the amount of product C , $(F_A + F_B)c_C$, multiplied by the yield factor $(F_A + F_B)c_C/F_Ac_{A_{in}}$. Upper bounds are defined for the amount of heat produced by the reactions and the total flow (see Table 3.3). The optimization can be formulated mathematically as:

$$\max_{F_A, F_B} \phi := \frac{(F_A + F_B)^2 c_C^2}{F_A c_{A_{in}}} \quad (3.34)$$

s.t. model equations (3.29)-(3.33)

$$G_1 := q_r - q_{r,\max} \leq 0$$

$$G_2 := F_A + F_B - F_{\max} \leq 0.$$

The optimal feed rates, the values of the constrained quantities, and the cost function for $\bar{k}_1 = 0.3, 0.75$ and $1.5 \frac{1}{\text{mol}\cdot\text{h}}$ are given in Table

⁴ Note that this value is different from the one used in [79].

3.4. Notice that the set of active constraints in the optimal solution changes with the value of \bar{k}_1 .

Table 3.4. Optimal solutions for various values of the plant parameter \bar{k}_1

\bar{k}_1	F_A^*	F_B^*	$\frac{q_r}{q_{r,\max}}$	$\frac{F_A+F_B}{F_{\max}}$	ϕ
0.30	8.21	13.79	0.887	1.000	8.05
0.75	8.17	13.83	1.000	1.000	11.16
1.50	7.61	13.05	1.000	0.940	12.30

Accuracy of the Constraint-Adaptation Algorithm.

Since the constraint G_2 is not affected by the uncertainty, only the constraint G_1 requires adaptation. This subsection investigates the accuracy of the constraint-adaptation algorithm upon convergence and in the absence of measurement noise and process disturbances (ideal case).

The scaled constrained quantities $q_r/q_{r,\max}$ and $(F_A+F_B)/F_{\max}$ are represented in Figure 3.5 for values of \bar{k}_1 (plant) varying in the range 0.3 to 1.5 $\frac{1}{\text{mol}\cdot\text{h}}$. Note that the constrained quantities obtained with the constraint-adaptation algorithm (thick lines) follow closely those of the true optimal solution (thin lines). But, although the proposed algorithm guarantees feasible operation upon convergence irrespective of the value of \bar{k}_1 , it fails to detect the correct active set in the vicinity of those operating points where the active set changes (i.e., $\bar{k}_1 \approx 0.65$ and $\bar{k}_1 \approx 0.8$). This deficiency results from the error introduced by the process model in the evaluation of the sensitivities with respect to F_A and F_B of both the cost function and the process-dependent constraint.

Figure 3.6 shows the performance loss

$$\Delta\phi := \frac{\phi(\mathbf{u}_p^*) - \phi(\mathbf{u}_\infty^*(k_1))}{\phi(\mathbf{u}_p^*)},$$

where $\phi(\mathbf{u}_p^*)$ denotes the true optimal cost, and $\phi(\mathbf{u}_\infty^*(k_1))$ the optimal cost obtained upon convergence of the constraint-adaptation algorithm. For example, using the model parameter $k_1 = 1.5$, $\Delta\phi$ is equal

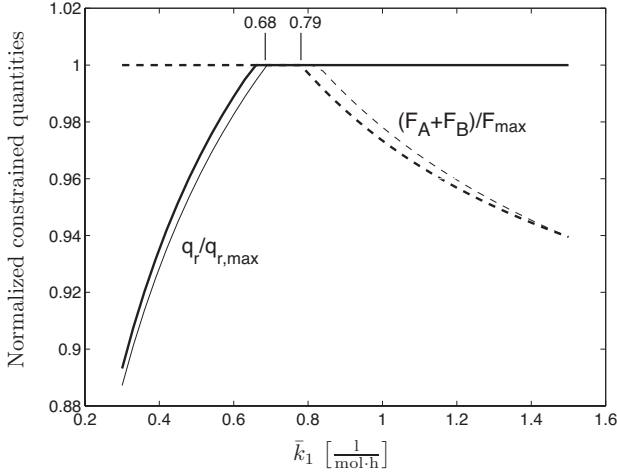


Fig. 3.5. Optimal values of the constrained quantities $q_r/q_{r,\max}$ and $(F_A + F_B)/F_{\max}$ versus the plant parameter \bar{k}_1 , for $k_1 = 1.5 \frac{1}{\text{mol}\cdot\text{h}}$. Thick lines: Constraint-adaptation solution; Thin lines: True optimal solution.

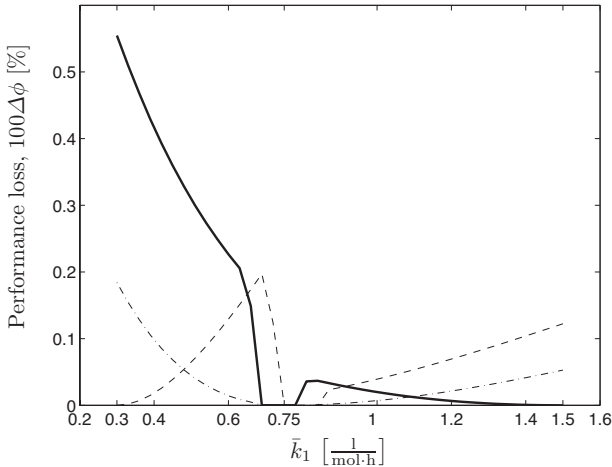


Fig. 3.6. Performance loss of the constraint-adaptation solution in function of the plant parameter \bar{k}_1 for three values of the model parameter k_1 : thick line: $k_1 = 1.5 \frac{1}{\text{mol}\cdot\text{h}}$; thin dot-dashed line: $k_1 = 0.75 \frac{1}{\text{mol}\cdot\text{h}}$; thin dashed line: $k_1 = 0.3 \frac{1}{\text{mol}\cdot\text{h}}$.

to zero for $\bar{k}_1 = 1.5$, for there is no model mismatch in this case. Interestingly enough, $\Delta\phi$ is also equal to zero when the two constraints are active and the adaptation scheme provides the correct active set; this situation occurs for \bar{k}_1 in the range $0.68 < \bar{k}_1 < 0.79$ (see Figure 3.5). Overall, the performance loss remains lower than 0.6% for any value of \bar{k}_1 in the range 0.3 to $1.5 \frac{1}{\text{mol}\cdot\text{h}}$, and is even lower (less than 0.2%) when the model parameter k_1 is chosen as 0.3 or $0.75 \frac{1}{\text{mol}\cdot\text{h}}$. These results demonstrate that the performance loss remains limited, despite the error made in the detection of the active set for some scenarios.

Constraint-Adaptation Results.

In this subsection, we take a closer look at the iterations produced by the constraint-adaptation algorithm. Two scenarios that correspond to different sets of active constraints at the optimum are considered. In either scenario, the process model with $k_1 = 1.5 \frac{1}{\text{mol}\cdot\text{h}}$ is chosen. Note also that the adaptation is started with a large (conservative) initial constraint modifier $\varepsilon_0 = 1.5 \times 10^5 \frac{\text{J}}{\text{h}}$, and the filter parameter for ε is taken as $b := 1$ (no filtering).

To depart from the ideal case of the previous subsection, Gaussian noise with standard deviation of $1800 \frac{\text{J}}{\text{h}}$ is added to the estimate of q_r . In response to this, a back-off is introduced to ensure that the heat production constraint is satisfied, i.e., $q_{r,\max} = 9.9 \times 10^5 \frac{\text{J}}{\text{h}}$. Two scenarios, corresponding to a plant (simulated reality) with $\bar{k}_1 = 0.75 \frac{1}{\text{mol}\cdot\text{h}}$ and $\bar{k}_1 = 0.3 \frac{1}{\text{mol}\cdot\text{h}}$ are considered.

Scenario A: Plant with $\bar{k}_1 = 0.75 \frac{1}{\text{mol}\cdot\text{h}}$.

The evolution of the constraint modifier ε with the RTO period is shown in Figure 3.7 (thick line). A negative value of ε indicates that the heat production is overestimated by the model, which is consistent with the larger value of \bar{k}_1 chosen for the plant. Note also that the convergence is very fast in this case, as the region where the adaptation is within the noise level is reached after two RTO periods only. The corresponding constrained quantities ($F_A + F_B$) and q_r are represented in Figure 3.8. Observe that only the heat production constraint is active in this scenario, and that the chosen back-off ensures that the maximum value of $10^6 \frac{\text{J}}{\text{h}}$ does not get exceeded despite measurement noise. On the other hand, the feed rate constraint remains inactive, although its value gets close to the maximum feed rate. Finally, the

evolution of the cost function ϕ is shown in Figure 3.9 (thick line). The converged cost value is close to 11, i.e., within a few percent of the ideal cost given in Table 3.4, despite the performance loss induced by backing-off the heat production constraint.

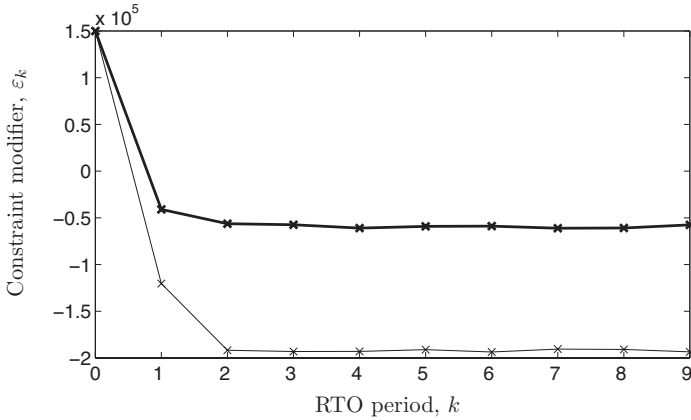


Fig. 3.7. Evolution of the constraint modifier ε . Thick line: Scenario A; Thin line: Scenario B.

Scenario B: Plant with $\bar{k}_1 = 0.3 \frac{l}{mol h}$.

The evolution of the constraint modifier ε , the constrained quantities $(F_A + F_B)$ and q_r , and the objective function ϕ , with the RTO iteration is shown as thin lines in Figures 3.7, 3.8 and 3.9, respectively. It is seen from Figure 3.7 that the constraint modifier is larger in this scenario than in the previous one, as the process model is even farther from the plant. Moreover, Figure 3.8 shows that only the feed rate constraints gets active, while the heat production remains inactive. Hence, the optimal inputs are unaffected by the measurement noise. It takes a single RTO period for the constraint-adaptation algorithm to detect the correct active set in this case. Finally, it is seen from Figure 3.9 that the converged cost value of about 8 is very close to the ideal cost reported in Table 3.4, in spite of the large model mismatch.

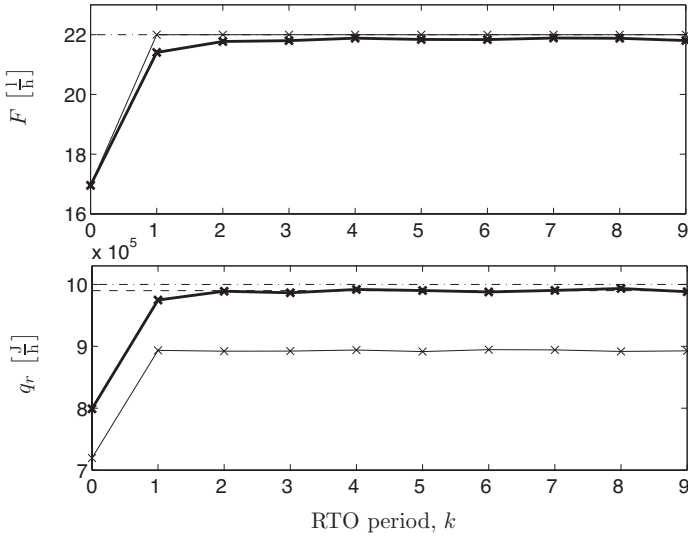


Fig. 3.8. Evolution of the constrained quantities. Thick lines: Scenario A; Thin lines: Scenario B; Thin dot-dashed lines: Upper bounds on the constrained quantities; Thin dashed line: Constraint back-off.

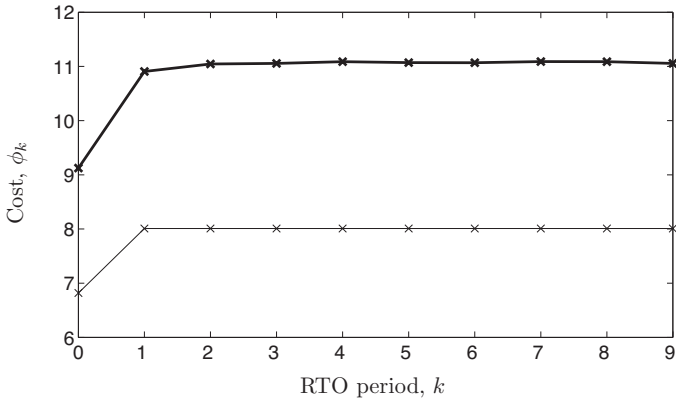


Fig. 3.9. Evolution of the cost function. Thick lines: Scenario A; Thin lines: Scenario B.

3.3 Combination of Constraint Adaptation with Constraint Control

Two types of transient behaviors can be distinguished in RTO systems: (i) the dynamic regime of the controlled plant between two successive steady-state operations, and (ii) the transient produced by the iterations of the RTO algorithm before convergence. Most RTO algorithms do not ensure feasibility during these transient periods, thus calling for a conservative implementation with significant constraint backoffs and reduced movements of the operating point between successive RTO periods [83]. Constraint violations during both types of transients can be dealt with by controlling the constraints that are active at the optimum [40, 56, 83]. The set of active constraints might change due to plant-model mismatch, process disturbances and changing operating conditions, thus resulting in different constraint controllers.

Constraint adaptation guarantees that a feasible operating point for the plant will be reached upon convergence. However, the iterates may follow an infeasible path, even if the adaptation is started at a feasible point. Process disturbances and changes in the operating conditions may also result in violations of the constraints. The process control system should then be designed to implement the RTO results while taking care not to violate the constraints. MPC is a natural choice for this task because of its ability to handle large multivariable control problems and to anticipate the effect of the constraints. The approach described in this section for integrating constraint adaptation at the RTO level with MPC at the process control level (see Figure 1.1) places high emphasis on how constraints are handled. A term is added to the cost function of the MPC problem in order to assign the optimal values provided by the RTO level to the residual degrees of freedom of the control problem.

In this section, equality constraints are explicitly included in the optimization problem, since we want to emphasize the fact that these constraints are controlled in the MPC constraint controller. The inclusion of these constraints poses no conceptual difficulty, as they behave in a constraint adaptation scheme similar to inequality constraints that are always active.

3.3.1 Constraint-Adaptation Scheme for Combination with MPC

Problem Formulation

Including equality constraints, the steady-state optimization problem for the plant can be formulated as follows:

$$\begin{aligned}
 \min_{\mathbf{u}} \quad & \Phi_p(\mathbf{u}) = \phi(\mathbf{u}, \mathbf{y}_p(\mathbf{u})) & (3.35) \\
 \text{s.t.} \quad & \mathbf{H}_p(\mathbf{u}) := \bar{\mathbf{h}}(\mathbf{u}, \mathbf{y}_p(\mathbf{u})) = \mathbf{H}^S \\
 & \bar{\mathbf{G}}_p(\mathbf{u}) = \bar{\mathbf{g}}(\mathbf{u}, \mathbf{y}_p(\mathbf{u})) \leq \bar{\mathbf{G}}^U \\
 & \mathbf{u}^L \leq \mathbf{u} \leq \mathbf{u}^U,
 \end{aligned}$$

where $\bar{\mathbf{h}} \in \mathbb{R}^{n_h}$ are the equality constrained functions for which \mathbf{H}^S are the setpoint values; and $\bar{\mathbf{g}} \in \mathbb{R}^{n_g}$ are the inequality constrained functions for which $\bar{\mathbf{G}}^U$ are the upper bounds. It is assumed that ϕ , $\bar{\mathbf{h}}$ and $\bar{\mathbf{g}}$ are known functions of \mathbf{u} and \mathbf{y} , i.e., they can be evaluated directly from the measurements.

The model-based optimization problem, on the other hand, is given by:

$$\begin{aligned}
 \min_{\mathbf{u}} \quad & \Phi(\mathbf{u}, \boldsymbol{\theta}) = \phi(\mathbf{u}, \mathbf{y}(\mathbf{u}, \boldsymbol{\theta})) & (3.36) \\
 \text{s.t.} \quad & \mathbf{H}(\mathbf{u}, \boldsymbol{\theta}) := \bar{\mathbf{h}}(\mathbf{u}, \mathbf{y}(\mathbf{u}, \boldsymbol{\theta})) = \mathbf{H}^S \\
 & \bar{\mathbf{G}}(\mathbf{u}, \boldsymbol{\theta}) = \bar{\mathbf{g}}(\mathbf{u}, \mathbf{y}(\mathbf{u}, \boldsymbol{\theta})) \leq \bar{\mathbf{G}}^U \\
 & \mathbf{u}^L \leq \mathbf{u} \leq \mathbf{u}^U.
 \end{aligned}$$

Constraint-Adaptation Scheme

For implementation with MPC, the constraints are corrected at the iteration k as follows:

$$\mathbf{H}(\mathbf{u}, \boldsymbol{\theta}) + \gamma_k^{\mathbf{H}} = \mathbf{H}^S, \quad (3.37)$$

$$\bar{\mathbf{G}}(\mathbf{u}, \boldsymbol{\theta}) + \gamma_k^{\mathbf{G}} \leq \bar{\mathbf{G}}^U, \quad (3.38)$$

with

$$\gamma_k^{\mathbf{H}} := \mathbf{H}_p(\mathbf{u}_k) - \mathbf{H}(\mathbf{u}_k, \boldsymbol{\theta}), \quad (3.39)$$

$$\gamma_k^{\mathbf{G}} = \bar{\mathbf{G}}_p(\mathbf{u}_k) - \bar{\mathbf{G}}(\mathbf{u}_k, \boldsymbol{\theta}), \quad (3.40)$$

$$\bar{G}_{i,k}^{\mathbf{U}} = \begin{cases} \bar{G}_{p,i}(\mathbf{u}_k) + b_i (\bar{G}_i^{\mathbf{U}} - \bar{G}_{p,i}(\mathbf{u}_k)), & \text{if } \bar{G}_{p,i}(\mathbf{u}_k) \leq \bar{G}_i^{\mathbf{U}} \\ \bar{G}_i^{\mathbf{U}}, & \text{otherwise.} \end{cases} \quad (3.41)$$

$i = 1, \dots, n_g$

Here, direct adaptation of constraint bounds is used to modify the inequality constraints \mathbf{G} only when the adaptation takes place inside the feasible region, as expressed by (3.41). As will be discussed in Subsection 3.3.2, the values of $\bar{\mathbf{G}}_k^{\mathbf{U}}$ may be selected as setpoints of the constraint controller. Hence, (3.41) prevents a setpoint value greater than $\bar{\mathbf{G}}^{\mathbf{U}}$ from being selected. This approach permits to gradually move the setpoints $\bar{\mathbf{G}}_k^{\mathbf{U}}$ that will be passed to the MPC constraint controller. The setpoints $\bar{\mathbf{G}}_k^{\mathbf{U}}$ for the active constraints reach the actual constraint bounds $\bar{\mathbf{G}}^{\mathbf{U}}$ upon convergence. In order to account for measurement noise, a back-off can be applied to $\bar{\mathbf{G}}^{\mathbf{U}}$. Notice that it is also possible to move in a similar way the input bounds.

At the k th iteration, the next optimal input values are computed:

$$\begin{aligned} \mathbf{u}_{k+1}^* &\in \arg \min_{\mathbf{u}} \quad \Phi(\mathbf{u}, \boldsymbol{\theta}) & (3.42) \\ \text{s.t.} \quad &\mathbf{H}(\mathbf{u}, \boldsymbol{\theta}) + \gamma_k^{\mathbf{H}} = \mathbf{H}^{\mathbf{S}} \\ &\bar{\mathbf{G}}(\mathbf{u}, \boldsymbol{\theta}) + \gamma_k^{\mathbf{G}} \leq \bar{\mathbf{G}}_k^{\mathbf{U}} \\ &\mathbf{u}^{\mathbf{L}} \leq \mathbf{u} \leq \mathbf{u}^{\mathbf{U}}. \end{aligned}$$

When constraint adaptation is applied alone, the new operating point is obtained by applying the optimal inputs \mathbf{u}_{k+1}^* directly to the plant. However, in combining constraint adaptation with MPC, the inputs are determined by the MPC controller, and the values \mathbf{u}_{k+1} to be used in the next RTO iteration correspond to the input values reached by the controlled plant at steady state.

Illustrative Example

The approach for integrating constraint adaptation with MPC assumes that all the constrained variables can be measured or estimated on-line with a sampling period much smaller than the natural response time of the controlled plant.

Table 3.5. Values of the parameters θ in Problem (3.4) for the model and the plant (simulated reality).

	θ_1	θ_2	θ_3	θ_4
Plant	0.4	0.8	-1.8	1.9
Model	0.9	0.4	-2.0	1.4

Direct adaptation of the constraint bounds is especially useful when RTO is started from a conservative (feasible) operating point and one wants to approach the constraints gradually from within the feasible region. This is illustrated in the next example.

Example 3.4 Consider the following QP problem:

$$\begin{aligned} \min \quad & \Phi(\mathbf{u}, \boldsymbol{\theta}) := (u_1 - 1)^2 + (u_2 - 1)^2 & (3.43) \\ \text{s.t.} \quad & G_1 := \theta_1 + \theta_2 u_1 - u_2 \leq 0 \\ & G_2 := \theta_3 + \theta_4 u_1 + u_2 \leq 0, \end{aligned}$$

with two decision variables $\mathbf{u} = [u_1 \ u_2]^\top$, four model parameters $\boldsymbol{\theta} = [\theta_1 \ \dots \ \theta_4]^\top$, and two uncertain constraints G_1 and G_2 . The parameter values for the model and the plant (simulated reality) are reported in Table 3.5. Note that the operating point determined from the model, without constraint adaptation, leads to constraint violation.

In this simple QP problem, an ideal constraint controller is assumed. The objective here is to illustrate the effect of constraint control on the feasibility of the successive steady-state operating points. Both constraints are active at the optimum for both the plant and the model. The constraint-adaptation algorithm is applied with and without constraint control, starting from $\mathbf{u}_0 = [0 \ 1.4]^\top$ and with a diagonal gain matrix $\mathbf{B} = b \mathbf{I}_2$ with $b \in (0, 1]$. The results obtained with $b = 0.7$ are shown in Figure 3.10. It can be seen that, without constraint control, the iterates converge by following an infeasible path (left plot). In fact, the iterates can be shown to follow an infeasible path for any value of $b \in (0, 1]$; the constraint violation is reduced by decreasing the value of b , but this is at the expense of a slower convergence. With constraint control, on the other hand, the iterates converge without violating the constraints (right plot), irrespectively of the value of b . Both constraints are found to be active at the solution point of the optimization problem (3.42) for all iterations.

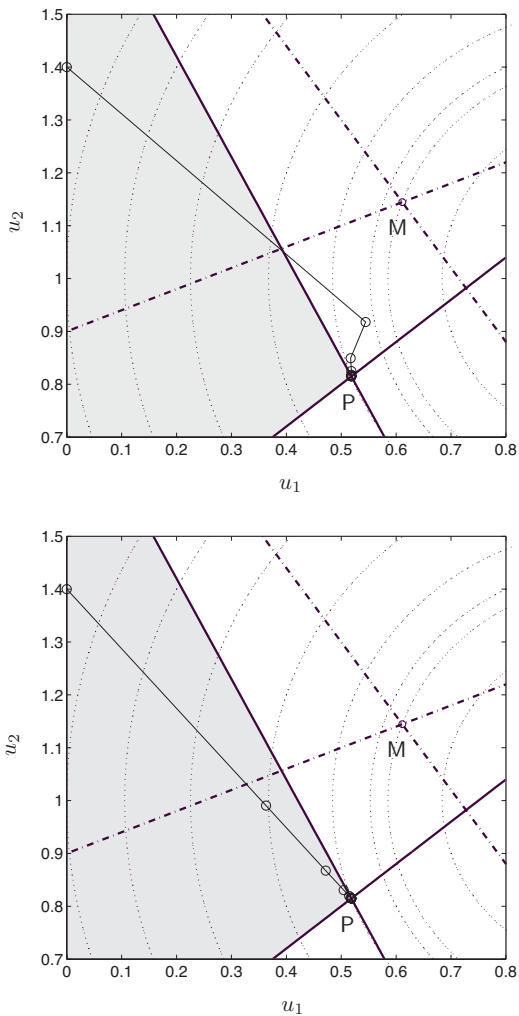


Fig. 3.10. Illustration of the proposed algorithm for Problem (3.43). **Top plot:** Without constraint control; **Bottom plot:** With constraint control; **Colored area:** feasible region; **Thick solid lines:** constraint bounds for the plant; **Thick dash-dotted lines:** constraint bounds predicted by the model without adaptation; **Dotted lines:** contours of the cost function; **Light solid lines:** iterates; **Point P:** optimum for the plant; **Point M:** optimum for the model without adaptation.

The design of the MPC constraint controller is described in the following subsection.

3.3.2 Enforcing Constraints via MPC

Constraint adaptation guarantees reaching a feasible operating point upon convergence, but it does not ensure feasibility neither at the successive steady-state operating points produced by the algorithm nor during the transient between two successive steady-state operating points. Constraint violations can be prevented by controlling the constraints. It is generally recognized that equality constraints as well as active inequality constraints should be controlled. An additional complication results from the fact that, in the presence of process disturbances and changing operating conditions, the set of active inequality constraints may change, thus requiring a change in the plant's control structure [40, 56].

The equality constraints \mathbf{H} are controlled at their setpoint values \mathbf{H}^S . Between the RTO iterations k and $k+1$, the subset $(\bar{\mathbf{G}})_{r,k} \in \mathbb{R}^{(n_g)_{r,k}}$ of the inequality constraints $\bar{\mathbf{G}}$ are controlled at the setpoints $(\bar{\mathbf{G}}^S)_{r,k} := (\bar{\mathbf{G}})_{r,k}(\mathbf{u}_{k+1}^*, \boldsymbol{\theta}) + (\boldsymbol{\gamma}_k^{\mathbf{G}})_{r,k}$.⁵ The controlled constraints $(\bar{\mathbf{G}})_{r,k}$ can be selected as the active inequality constraints at the k th RTO iteration, thus leading to $(\bar{\mathbf{G}}^S)_{r,k} = (\bar{\mathbf{G}}_k^U)_{r,k}$.

Let us assume that at the k th RTO iteration, at the solution of Problem (3.42), there are n_k^L inputs at their lower bound, and n_k^U inputs at their upper bound. We define the matrix $\mathbf{P}_k^L \in \mathbb{R}^{n_k^L \times n_u}$ such that each row of \mathbf{P}_k^L has a 1 at the index number corresponding to an active lower input bound and zeros elsewhere. This way, $\mathbf{P}_k^L(\mathbf{P}_k^L)^T = \mathbf{I}_{n_k^L}$ and $(\mathbf{P}_k^L)^T \mathbf{P}_k^L \in \mathbb{R}^{n_u \times n_u}$ is a diagonal matrix whose diagonal elements corresponding to an active lower input bound are 1, and those corresponding to an inactive lower input bound are 0. Similar to \mathbf{P}_k^L , matrix $\mathbf{P}_k^U \in \mathbb{R}^{n_k^U \times n_u}$ is defined for the inputs that are at their upper bound.

This way, the controlled inequality constraints and input variables between the RTO iterations k and $k+1$ can be denoted collectively as:

⁵ The notation is involved and requires some explanation. Here, $\boldsymbol{\gamma}_k^{\mathbf{G}}$ are the correction terms given by (3.40), and $(\boldsymbol{\gamma}_k^{\mathbf{G}})_{r,k}$ is the subset of $\boldsymbol{\gamma}_k^{\mathbf{G}}$ corresponding to the controlled inequality constraints between the RTO iterations k and $k+1$. Notice that the controlled inequality constraints might change from one iteration to the other.

$$\mathbf{z}_{r,k} := \begin{bmatrix} (\bar{\mathbf{G}})_{r,k} \\ -\mathbf{P}_k^L \mathbf{u} \\ \mathbf{P}_k^U \mathbf{u} \end{bmatrix},$$

and the corresponding setpoints are:

$$\mathbf{z}_{r,k}^S := \begin{bmatrix} (\bar{\mathbf{G}}^S)_{r,k} \\ -\mathbf{P}_k^L \mathbf{u}^L \\ \mathbf{P}_k^U \mathbf{u}^U \end{bmatrix},$$

where $\mathbf{z}_{r,k}$, $\mathbf{z}_{r,k}^S \in \mathbb{R}^{(n_z)_k}$, with $(n_z)_k = (n_g)_{r,k} + n_k^L + n_k^U$.

Typically, the active constraints are selected as controlled variables; however, inactive constraints can also be selected as controlled variables as long as $(n_z)_k \leq (n_u - n_h)$ to avoid over-specification. In practice, input-output selection criteria should also guide the selection of the inequality constraints to be included in $\mathbf{z}_{r,k}$. For example, near steady-state collinearity between controlled constraints should be avoided, since this would make the controlled plant become sensitive to process disturbances.

The methodology for combining the constraint-adaptation scheme with MPC is presented using a step response model of truncation order n . This prediction model is used in dynamic matrix control (DMC), which is one of the original MPC formulations, and still one of the most popular MPC algorithms in industry [57]. However, other prediction models could be used as well. Using MPC, constraint control can be implemented between the RTO iterations k and $k+1$ by minimizing the quadratic objective function

$$J_k(\mathbf{u}(t)) = \sum_{l=1}^p \left\{ \left\| \mathbf{P}_l \left[\mathbf{H}(t+l|t) - \mathbf{H}^S \right] \right\|^2 + \left\| \mathbf{Q}_{l,k} \left[\mathbf{z}_{r,k}(t+l|t) - \mathbf{z}_{r,k}^S \right] \right\|^2 + \left\| \mathbf{R}_l \Delta \mathbf{u}(t+l-1) \right\|^2 \right\} \quad (3.44)$$

where \mathbf{P}_l is the weighting matrix on the equality constraints, \mathbf{H}^S the setpoint values for the equality constraints, $\mathbf{Q}_{l,k}$ the weighting matrix on the controlled inequality constraints, $\mathbf{z}_{r,k}^S$ the setpoint values for the inequality constraints, and \mathbf{R}_l the weighting matrix on the rate of change of the inputs.

At current time t , the behavior of the process over p future time steps is considered. MPC determines the next m input moves $\Delta \mathbf{u}(t+l|t) := \mathbf{u}(t+l|t) - \mathbf{u}(t+l-1|t)$, $l = 0, \dots, m-1$, with $m < p$ and $\Delta \mathbf{u}(t+$

$l|t) = \mathbf{0}$, $\forall l \geq m$. Only the first computed change in the manipulated variables is implemented, and at time $t+1$ the computation is repeated with the horizon moved by one time interval:

$$\Delta \mathbf{u}(t|t)^\top, \dots, \Delta \mathbf{u}(t+m-1|t)^\top \quad \min_{\mathbf{u}(t)} \quad J_k(\mathbf{u}(t)) \quad (3.45)$$

s.t.

$$\begin{bmatrix} \mathbf{H}(t+l|t) \\ \bar{\mathbf{G}}(t+l|t) \end{bmatrix} = \sum_{i=1}^l \mathbf{S}_i \Delta \mathbf{u}(t+l-i) + \sum_{i=l+1}^{n-1} \mathbf{S}_i \Delta \mathbf{u}(t+l-i) \quad (3.46)$$

$$+ \mathbf{S}_n \mathbf{u}(t+l-n) + \mathbf{d}(t+l|t), \quad l = 1, \dots, p,$$

$$\mathbf{d}(t+l|t) = \mathbf{d}(t|t)$$

$$= \begin{bmatrix} \mathbf{H}_p(t) \\ \mathbf{G}_p(t) \end{bmatrix} - \sum_{i=1}^{n-1} \mathbf{S}_i \Delta \mathbf{u}(t-i) - \mathbf{S}_n \mathbf{u}(t-n), \quad l = 1, \dots, p, \quad (3.47)$$

$$(\bar{\mathbf{G}})_{s,k}(t+l|t) \leq (\bar{\mathbf{G}}^U)_{s,k}, \quad l = 1, \dots, p, \quad (3.48)$$

$$\mathbf{u}(t+l|t) = \sum_{i=0}^l \Delta \mathbf{u}(t+i|t) + \mathbf{u}(t-1), \quad l = 0, \dots, m-1, \quad (3.49)$$

$$\mathbf{u}^L \leq \mathbf{u}(t+l|t) \leq \mathbf{u}^U, \quad l = 0, \dots, m-1, \quad (3.50)$$

$$\Delta \mathbf{u}(t+l|t) \leq \Delta \mathbf{u}^U, \quad -\Delta \mathbf{u}(t+l|t) \leq \Delta \mathbf{u}^U, \quad (3.51)$$

$$l = 0, \dots, m-1,$$

where $\mathbf{H}(t+l|t)$ and $\bar{\mathbf{G}}(t+l|t)$ are the model prediction of the equality and inequality constraints \mathbf{H} and $\bar{\mathbf{G}}$ at time $t+l$ based on information available at time t , respectively. This prediction involves on the right-hand side of (3.46) a first term including the effect of the present and future input moves, a second and third terms including the effect of the past input moves, and a fourth term $\mathbf{d}(t+l|t) \in \mathbb{R}^{n_h+n_g}$ that corresponds to the predicted constraint biases at time $t+l$ obtained at time t . These constraint biases are computed in (3.47) as the difference between the measured value of the constraints $\mathbf{H}_p(t)$ and $\mathbf{G}_p(t)$ and their model predictions. Note that $\mathbf{d}(t+l|t)$ is assumed to be equal to $\mathbf{d}(t|t)$ for all future times ($l \geq 0$). The matrices $\mathbf{S}_i \in \mathbb{R}^{(n_h+n_g) \times n_u}$, $i = 1, \dots, n$, contain the step responses of \mathbf{H} and $\bar{\mathbf{G}}$ (see e.g. [57]). $(\bar{\mathbf{G}})_{s,k} \in \mathbb{R}^{(n_g)_{s,k}}$ are all the inequality constraints $\bar{\mathbf{G}}$ that are not included in $(\bar{\mathbf{G}})_{r,k}$. That is, in order not to violate the uncontrolled constraints $(\bar{\mathbf{G}})_{s,k}$, these are included in the formulation of the MPC problem in equation (3.48), as has been proposed in [40]. Note that

$(n_g)_{r,k} + (n_g)_{s,k} = n_g$. Equation (3.50) are the input bounds, and (3.51) provides bounds on the input moves.

In order to avoid possible infeasibility problems due to the presence of output-dependent constraints in (3.48), it is recommended to use a soft-constraint approach [76]. This approach consists in adding an additional penalty term to the objective function of the MPC problem, that penalizes a measure of constraint violation. With $(\boldsymbol{\tau})_{s,k}(t+l|t)$ denoting the predicted constraint violations, the term $\|\mathbf{W}_{l,k}(\boldsymbol{\tau})_{s,k}(t+l|t)\|^2 + \mathbf{w}_{l,k}^\top(\boldsymbol{\tau})_{s,k}(t+l|t)$ is added to the summation of the objective function of the MPC problem in (3.44). The constraints in (3.48) are softened as $(\mathbf{G})_{s,k}(t+l|t) \leq (\mathbf{G}^U)_{s,k} + (\boldsymbol{\tau})_{s,k}(t+l|t)$, with $(\boldsymbol{\tau})_{s,k}(t+l|t) \geq \mathbf{0}$ and the constraint violations $(\boldsymbol{\tau})_{s,k}(t+l|t)$ are included as decision variables in the MPC problem (3.45).

For the case of a non-square control problem with more inputs than controlled variables, i.e., $n_u > (n_z)_k + n_h$, an additional quadratic term is added to the MPC objective function to exploit the additional degrees of freedom towards optimality:

$$\begin{aligned} J_k(\mathbf{u}(t)) = \sum_{l=1}^p \left\{ \right. & \left\| \mathbf{P}_l \left[\mathbf{H}(t+l|t) - \mathbf{H}^S \right] \right\|^2 \\ & + \left\| \mathbf{Q}_{l,k} \left[\mathbf{z}_{r,k}(t+l|t) - \mathbf{z}_{r,k}^S \right] \right\|^2 + \left\| \mathbf{R}_l \Delta \mathbf{u}(t+l-1) \right\|^2 \\ & \left. + \left\| \mathbf{C}_l \boldsymbol{\xi}_k(t+l-1) \right\|^2 \right\} \end{aligned} \quad (3.52)$$

with

$$\boldsymbol{\xi}_k(t+l) = \mathbf{A}_{k+1}^\top \mathbf{u}(t+l|t) - \mathbf{A}_{k+1}^\top \mathbf{u}_{k+1}^*, \quad l = 0, \dots, m-1. \quad (3.53)$$

The columns of the matrix $\mathbf{A}_{k+1} \in \mathbb{R}^{n_u \times (n_u - (n_z)_k - n_h)}$ correspond to directions in the input space. The vector $\boldsymbol{\xi}_k \in \mathbb{R}^{n_u - (n_z)_k - n_h}$ is the difference between the inputs along these directions and their optimal values. The additional term $\|\mathbf{C}_l \boldsymbol{\xi}_k(t+l-1)\|^2$ in (3.52) allows controlling the inputs to their optimal values along the directions given by \mathbf{A}_{k+1} , thus addressing the $(n_u - (n_z)_k - n_h)$ residual degrees of freedom in the control problem. \mathbf{C}_l is the weighting matrix for $\boldsymbol{\xi}_k$. \mathbf{A}_{k+1} can be selected from information given by the steady-state model used in the RTO optimization [61]. A good choice is to select directions that are tangential to the constraints \mathbf{H} and $\mathbf{z}_{r,k}$ at \mathbf{u}_{k+1}^* . Assuming that the gradients of the controlled constraints \mathbf{H} and $\mathbf{z}_{r,k}$ are linearly independent at \mathbf{u}_{k+1}^* , the input space can be split into the

$((n_z)_k + n_h)$ -dimensional subspace of *constraint-seeking directions* and the $(n_u - (n_z)_k - n_h)$ -dimensional subspace of *sensitivity-seeking directions*. These subspaces are spanned by the columns of the orthogonal matrices \mathbf{V}_{k+1}^c and \mathbf{V}_{k+1}^s , respectively, as obtained by singular-value decomposition (SVD):

$$\begin{aligned} \left[\begin{array}{c} \frac{\partial \mathbf{H}}{\partial \mathbf{u}} \\ \frac{\partial \mathbf{z}_{r,k}}{\partial \mathbf{u}} \end{array} \right]_{(\mathbf{u}_{k+1}^*)} &:= \mathcal{G}(\mathbf{u}_{k+1}^*) \\ &= [\mathbf{U}_{k+1}^c \ \mathbf{U}_{k+1}^s] [\Sigma_{k+1}^c \ \mathbf{0}] [\mathbf{V}_{k+1}^c \ \mathbf{V}_{k+1}^s]^\top \end{aligned}$$

The $(n_u - (n_z)_k - n_h)$ columns of \mathbf{V}_{k+1}^s form an orthonormal basis of the null space of $\mathcal{G}(\mathbf{u}_{k+1}^*)$, and are therefore tangent to the controlled equality and inequality constraint contours. The matrix \mathbf{A}_{k+1} in (3.53) can then be selected as $\mathbf{A}_{k+1} = \mathbf{V}_{k+1}^s$. Much insight on why this choice seems attractive can be gained by visualizing the situation for the simplified example given below.

Example 3.5 Consider the following constraint adaptation problem:

$$\begin{aligned} \mathbf{u}_{k+1}^* &= \arg \min_{\mathbf{u}} \Phi(\mathbf{u}, \boldsymbol{\theta}) \\ \text{s.t.} \quad &\bar{G}(\mathbf{u}, \boldsymbol{\theta}) + \gamma_k^G \leq \bar{G}_k^U \end{aligned} \quad (3.54)$$

where the input \mathbf{u} has two components u_1 and u_2 , and there is a single constrained quantity \bar{G} to be adapted using the adaptation given by (3.38), (3.40) and (3.41). This constraint is assumed to be active at the optimal solution. Figure 3.11 illustrates the case where, at the current operating point \mathbf{u}_k , Problem (3.54) is solved for \mathbf{u}_{k+1}^* . Since at the solution the constraint is active, the cost function of the MPC problem is chosen as follows:

$$\begin{aligned} J_k(\mathbf{u}(t)) &= \sum_{l=1}^p \left\{ \|q_l [\bar{G}(t+l|t) - \bar{G}_k^U]\|^2 + \|\mathbf{R}_l \Delta \mathbf{u}(t+l-1)\|^2 \right. \\ &\quad \left. + \|c_l \xi_k(t+l-1)\|^2 \right\} \end{aligned}$$

Prior to convergence of constraint adaptation, the actual value of the constraint for the plant $\bar{G}_p(\mathbf{u}_{k+1}^*)$ is different from its setpoint value $\bar{G}_k^U = \bar{G}(\mathbf{u}_{k+1}^*, \boldsymbol{\theta}) + \gamma_k^G$. The MPC controller will therefore try to track the setpoint value \bar{G}_k^U by taking the operation somewhere on the constraint contour for the plant given by $\bar{G}_p(\mathbf{u}) = \bar{G}_k^U$. In order to provide an optimal value to the extra degree of freedom, one possibility

is to control one of the input components to its optimal value. The case where u_2 is controlled to its optimal value is obtained by selecting $\xi_k(t+l) = u_2(t+l|t) - u_{2,k+1}^*$, $l = 0, \dots, m-1$. Consider the situation represented in the top plot of Figure 3.11. If \bar{G} and u_2 are controlled, the MPC controller would drive the operation to the point \mathbf{u}_{k+1}^a , e.g., by following the trajectory **a**. A better option for this example would be to control u_1 at its optimal value instead of u_2 . Alternatively, it is possible to select a convenient linear combination of the inputs by controlling the inputs at their optimal value along the sensitivity-seeking direction \mathbf{V}_{k+1}^s . This is the case when ξ_k is selected as in (3.53) with $\mathbf{A}_{k+1} = \mathbf{V}_{k+1}^s$. In this case, the MPC controller would bring the operation to the point \mathbf{u}_{k+1}^b , e.g., by following the trajectory **b**. Notice that the cost at \mathbf{u}_{k+1}^b is lower than that at \mathbf{u}_{k+1}^a .

A different situation is represented in the bottom plot of Figure 3.11. In this case, if \bar{G} and u_2 are controlled, the MPC controller would find some compromise between meeting the setpoint value for the constraint $G_p(\mathbf{u}) = \bar{G}_k^U$ and meeting the desired value of $u_2 = u_{2,k+1}^*$. This is so because $\bar{G}_k(\mathbf{u}) = \bar{G}_k^U$ does not intersect $u_2 = u_{2,k+1}^*$. At this compromise point, the constraint on \bar{G} could become violated for the plant. On the other hand, if the inputs are controlled to their optimal value along \mathbf{V}_{k+1}^s , the MPC controller would take the system to the point \mathbf{u}_{k+1} , e.g., by following the trajectory **c**.

Notice that the selection of \mathbf{A}_{k+1} as an orthonormal basis of the null space of $\mathcal{G}(\mathbf{u}_{k+1}^*)$ is a convenient choice, but not the unique choice. Other choices can be found that are appropriate for a particular application, as will be the case in the forthcoming application in Subsection 3.3.3.

3.3.3 Case Study: Planar Solid Oxide Fuel Cell System

Given the prohibitive cost of non-renewable energy sources in today's scenario, fuel cells are intensively investigated as alternative power sources for a broad scope of applications. Solid oxide fuel cells (SOFCs) are energy conversion devices that produce electrical energy by the reaction of a fuel with an oxidant. 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 [44, 95]. Due to changes in the power demand during operation,

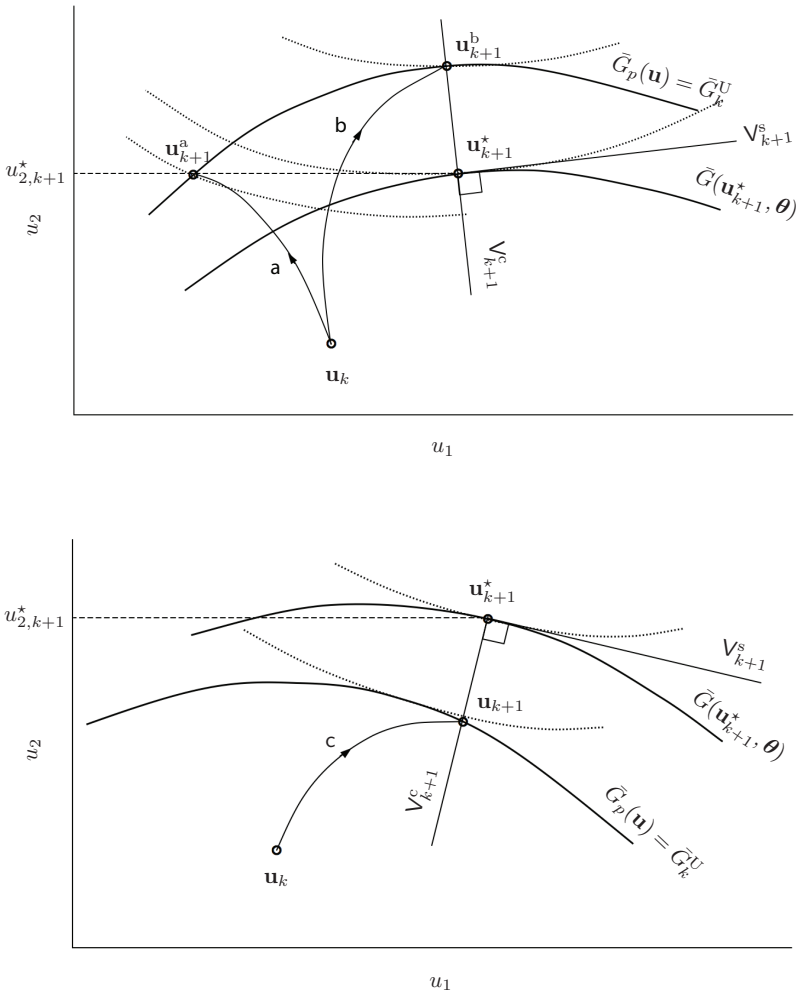


Fig. 3.11. Sketch of the integration between constraint adaptation and MPC using the sensitivity-seeking directions V^s . Dotted lines: contours of the cost function.

but also due to external perturbations affecting the SOFC system, the set of optimal operating conditions will vary with time continuously. Hence, there is a need for real-time optimization, i.e., repeated adjustment of the operating variables (e.g., flow rates, temperature) to maximize the performance (e.g., power output, efficiency) of the fuel cell.

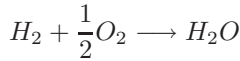
Different approaches have been proposed in the literature for controlling fuel cells. Aguiar et al. [1] discussed the use of PID feedback control in the presence of power load changes. For the case of a proton exchange membrane (PEM) fuel cell, Golbert and Lewin [43, 44] used a nonlinear MPC scheme with a target function that attempts to simultaneously track changes in the power setpoint and maximize efficiency. Recently, Zhang et al. [95] applied nonlinear MPC to a planar SOFC. However, these latter authors consider a square control problem, i.e., without residual degrees of freedom available for optimization. Several other control strategies for fuel cells have also been reported in the literature [47, 66, 86].

This subsection considers the RTO of a SOFC stack. An optimization problem maximizing the efficiency subject to operating constraints is defined. Due to inevitable model inaccuracies, the open-loop implementation of optimal inputs evaluated off-line may be suboptimal, or worse, infeasible. Infeasibility can be avoided by controlling the constrained quantities. However, the constraints that determine optimal operation might switch with varying power demand, thus requiring a change in the regulator structure.

Constraint adaptation guarantees reaching a feasible operating point upon convergence, that is, a point that satisfies all the constraints. However, the iterates may follow an infeasible path, even when adaptation is started from within the feasible region. Process disturbances and changes in the power demand may also result in constraint violation. Hence, MPC should implement the RTO results while avoiding constraint violations.

Description of the SOFC System

The system simulated is schematically represented in Figure 3.12. It comprises a 5-cell S-design planar SOFC stack operating in an electrically heated furnace [26]. The stack is fueled with 3% humidified H_2 . The only reaction taking place is the electrochemical oxidation of H_2 , for which the overall reaction is:



The furnace temperature is constant at 780 °C. The gas temperatures at the entrance of the stack are also considered constant at 750 °C. A blower outside the furnace delivers air for the cathode, whereas the fuel is provided directly at the desired pressure and flow rate.

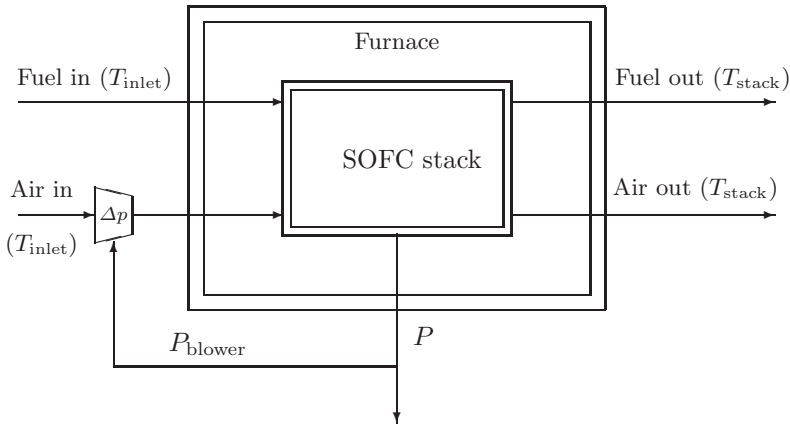


Fig. 3.12. Schematic of SOFC stack and furnace

A lumped model is used in this simulation study, as it captures the fundamental behavior of the SOFC while providing a good trade-off between accuracy and fast computation. The model has been validated from more detailed models developed at the Laboratoire d'Énergétique Industrielle (LENI) at EPFL, and it corresponds to SOFC stacks typically running at LENI's facilities [68, 87]. This lumped model considers a homogeneous temperature throughout the whole stack. The main equations of the model, which comprises energy equations, mass balances and electrochemical balances at the anode and cathode, are given in Appendix A. The specific nomenclature related with the model is given in Table A.3, and the parameter values are given in Table A.2.

I-V Curve.

A plot of cell voltage and power density as a function of the current density (I-V curve) is shown in Figure 3.13 for fuel inlet flow rates of $10^{-3} \frac{\text{mol}}{\text{sec}}$ (or mass flow density of $6 \frac{\text{ml}}{\text{min cm}^2}$) and $1.2 \times 10^{-3} \frac{\text{mol}}{\text{sec}}$ ($7.2 \frac{\text{ml}}{\text{min cm}^2}$), and an excess air ratio $\lambda_{\text{air}} = 3$.

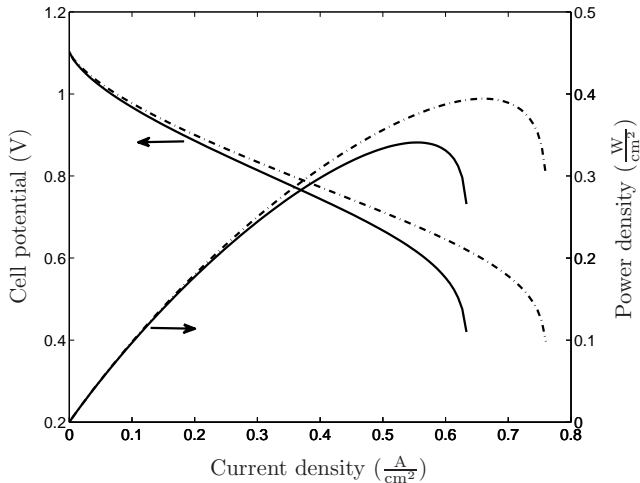


Fig. 3.13. Cell voltage and power density as a function of current density. Solid lines: $\dot{n}_{\text{fuel,in}} = 10^{-3} \frac{\text{mol}}{\text{sec}}$; Dot-dashed lines: $\dot{n}_{\text{fuel,in}} = 1.2 \times 10^{-3} \frac{\text{mol}}{\text{sec}}$.

At $\dot{n}_{\text{fuel,in}} = 10^{-3} \frac{\text{mol}}{\text{sec}}$, the limiting current density is $0.64 \frac{\text{A}}{\text{cm}^2}$. The maximum power density at which the cell can operate is $0.35 \frac{\text{W}}{\text{cm}^2}$. Increasing the current further, would result in a sharp dip in power due to increase in the overpotential losses. To deliver a higher power, it is necessary to increase the fuel inlet flow rate. At $\dot{n}_{\text{fuel,in}} = 1.2 \times 10^{-4} \frac{\text{mol}}{\text{sec}}$, it is possible to reach power densities of up to $0.4 \frac{\text{W}}{\text{cm}^2}$.

Formulation of the Optimization Problem

Input and Output Variables.

In the case of the SOFC system considered here, there are three degrees of freedom which can be specified as input variables: the molar fuel

flow rate at the anode, the molar air flow rate at the cathode, and the current:

$$\mathbf{u} = [\dot{n}_{\text{fuel,in}}, \dot{n}_{\text{air,in}}, I]^T \quad (3.55)$$

A similar choice of manipulated variables has been selected in [95] in the context of nonlinear MPC. The output variables are the stack temperature, cell potential, and power produced.

$$\mathbf{y} = [T_{\text{stack}}, U_{\text{cell}}, P]^T \quad (3.56)$$

Objective Function and Constraints.

The objective function to maximize is the electrical efficiency of the fuel cell for a given power demand, subject to operational constraints. Electrical efficiency is defined as the fraction of chemical power converted into useful power. Not all the power generated by the fuel cell is available for use. Due to pressure loss along the air channel, some power is used up internally by the blower to pump air. This power is the product of the pressure loss along the air channel and the volumetric flow rate of air. The electrical efficiency to maximize is thus,

$$\eta = \frac{P - P_{\text{blower}}}{\dot{n}_{\text{H}_2,\text{an,in}} LHV} = \frac{U_{\text{cell}} I N_{\text{cells}} - \Delta p \dot{Q}_{\text{air}}/\eta_{\text{blower}}}{\dot{n}_{\text{H}_2,\text{an,in}} LHV} \quad (3.57)$$

where the efficiency of the blower is $\eta_{\text{blower}} = 0.4$, and the pressure loss along the air channel, Δp , is proportional to the flow rate of air. LHV is the lower heating value of the fuel, which is the amount of heat released by combusting a mole of H_2 .

The fuel cell is operated under a number of inequality constraints including bounds on input and output variables (flow rates, cell potential, fuel utilization, stack temperature and current density). Constraints on the potential 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 [19, 87]. Operating at high current densities will cause material damage to the cell through excessive heating [51]. The low air-ratio limit is set to avoid high thermal gradients, while the high limit is due to system constraints. Current density is constrained to avoid degradation [51, 87]. The constraint bounds are given in Table 3.6. The setpoint value P_{el}^{S} for the produced power density is specified as an equality constraint.

The optimization problem can be formulated mathematically as:

Table 3.6. Values of constraint bounds (L: lower, U: upper)

$T_{\text{stack}}^{\text{L}}$	730	$^{\circ}\text{C}$	$T_{\text{stack}}^{\text{U}}$	800	$^{\circ}\text{C}$
$U_{\text{cell}}^{\text{L}}$	0.7	V	FU^{U}	70	%
$\lambda_{\text{air}}^{\text{L}}$	3		$\lambda_{\text{air}}^{\text{U}}$	7	
$\dot{n}_{\text{fuel,in}}^{\text{L}}$	5×10^{-4}	$\frac{\text{mol}}{\text{s}}$	i^{U}	0.6	$\frac{\text{A}}{\text{cm}^2}$

$$\begin{aligned}
 & \max_{\mathbf{u}} \quad \eta & (3.58) \\
 & \text{s.t.} \quad P_{\text{el}} = P_{\text{el}}^{\text{S}} \\
 & \quad T_{\text{stack}} \leq T_{\text{stack}}^{\text{U}}, & -T_{\text{stack}} \leq -T_{\text{stack}}^{\text{L}}, \\
 & \quad -U_{\text{cell}} \leq -U_{\text{cell}}^{\text{L}}, & FU \leq FU^{\text{U}}, \\
 & \quad \lambda_{\text{air}} \leq \lambda_{\text{air}}^{\text{U}}, & -\lambda_{\text{air}} \leq -\lambda_{\text{air}}^{\text{L}}, \\
 & \quad -\dot{n}_{\text{fuel,in}} \leq -\dot{n}_{\text{fuel,in}}^{\text{L}}, & i \leq i^{\text{U}}.
 \end{aligned}$$

Because the current density i and the power density P_{el} are not actually measured, they are considered to be proportional to the current I and the power P . Hence, the last constraint represents an input bound on the current.

Effect of Plant-Model Mismatch

In this simulation work, plant-model mismatch is considered by modifying certain model parameters. The modified parameters are given in Table 3.7, together with the corresponding values for the plant (simulated reality) and the nominal model.

Contour maps showing the objective function and the constraints at steady state as functions of the input variables for different power setpoints are presented in Figure 3.14 for the plant (simulated reality). These plots show the location of the plant optimum (point P) and the constraint bounds.

The objective function is directly proportional to the power output, and inversely dependant on the fuel flow rate. It also decreases with the air flow rate, as more power will be consumed by the blower. Notice that the set of active constraints at the optimum may change with the requested power densities. At the power density of $0.2 \frac{\text{W}}{\text{cm}^2}$, the optimum lies on the upper bound on fuel utilization (FU). The

Table 3.7. Values of the modified parameters for the plant and the nominal model

Parameter	Plant	Nominal model
$E_{\text{act,cath}} \left(\frac{\text{J}}{\text{mol}} \right)$	153260.5	150000
$k_{0,\text{cath}} \left(\frac{1}{\Omega \text{ m}^2} \right)$	4.103×10^{11}	4.5×10^{11}
$E_{\text{diss,cath}} \left(\frac{\text{J}}{\text{mol}} \right)$	2.473×10^{-19}	2.467×10^{-19}
$R_{0,\text{cath}} \left(\Omega \text{ m}^2 \right)$	9.2252×10^{-14}	10^{-13}

efficiency is about 40%. As the power setpoint is increased, the active constraint switches to the constraint on the cell potential, and it is not

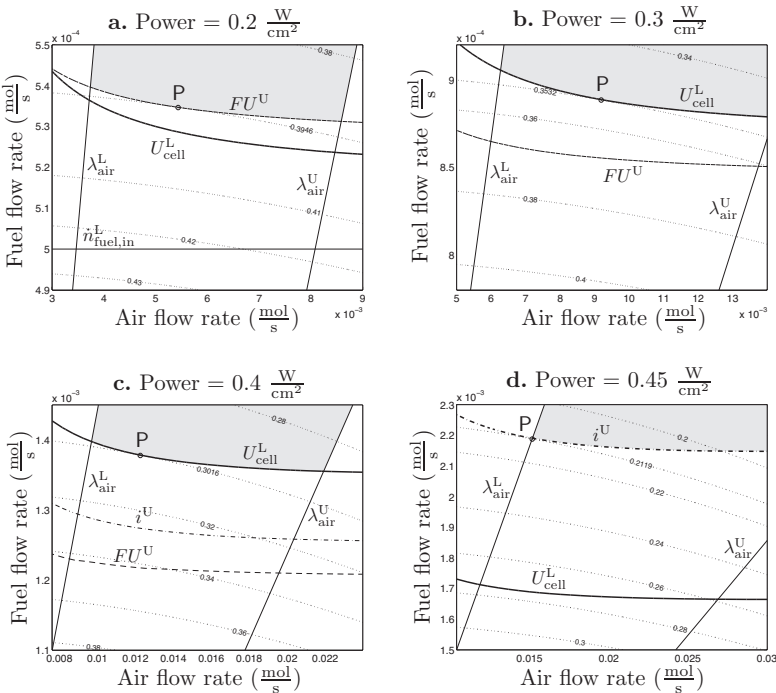


Fig. 3.14. Contour maps and operational constraints for the plant at steady state corresponding to different power setpoints. Colored area: feasible region; Dotted lines: contours of the cost function; Point P: optimum for the plant.

possible to reach the maximum FU . The optimum efficiency therefore drops. At a higher power density ($0.45 \frac{\text{W}}{\text{cm}^2}$), the active constraint is the one on current density, and the optimal operating point gives around 21% efficiency.

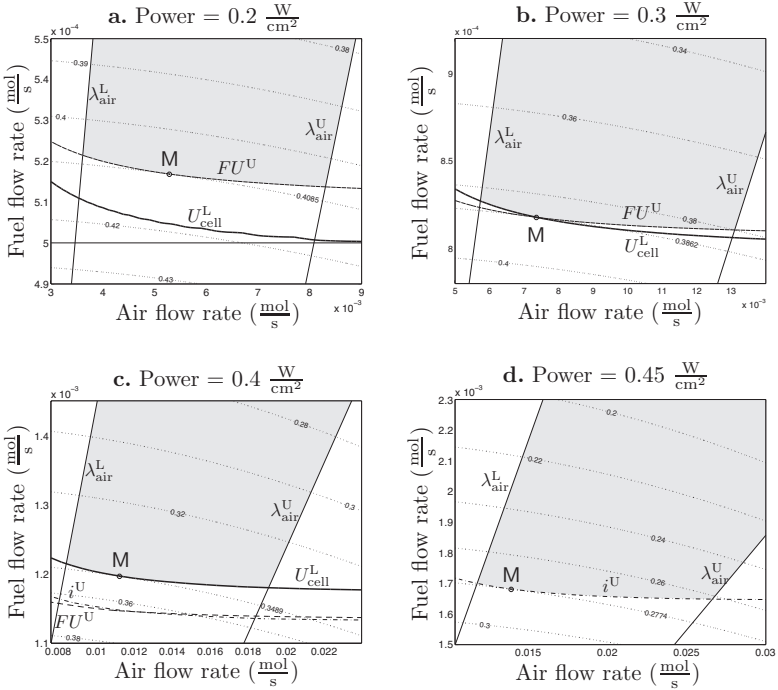


Fig. 3.15. Contour maps and operational constraints for the nominal model at steady-state corresponding to different power setpoints. Colored area: feasible region; Dotted lines: contours of the cost function; Point M: optimum for the model.

Similar contour maps can be drawn for the nominal model (Figure 3.15). Point M indicates the location of the model optimum. The constraints predicted by the model are different from those of the plant, and it is also possible that, for the same power setpoint, the set of active constraints at the optimum are different for the nominal model and the plant. For example, at the power density of $0.3 \frac{\text{W}}{\text{cm}^2}$, the active constraint for the plant is on the cell potential, whereas both the cell potential and the fuel utilization constraints are active for the model.

Even more different, at the power density of $0.45 \frac{\text{W}}{\text{cm}^2}$, the plant optimum is at the intersection of the lower bound on λ_{air} and the upper bound on the current density, whereas the lower bound on λ_{air} is not active for the nominal model.

Application of Constraint Adaptation Alone

The time constant of the fuel cell is around 40 s. A RTO period of 10 min is chosen, which leaves sufficient time for the system to reach steady-state after an input change. The constraint-adaptation scheme is applied using the parameter values of Table 3.7 for the plant and the nominal model. The nominal model corresponds to a steady-state model, i.e., with $\frac{\partial T_{\text{stack}}}{\partial t} = 0$ in (A.1). Figure 3.16 shows the response of some of the key variables. Initially, the plant is at steady-state with the power setpoint $P_{\text{el}}^{\text{S}} = 0.4 \frac{\text{W}}{\text{cm}^2}$ and the corresponding input $\mathbf{u}_0 = [19 \times 10^{-4}, 14 \times 10^{-3}, 26.00]^{\text{T}}$. Constraint adaptation is started at $t = 10$ min with $\mathbf{B} = \mathbf{I}_2$. Since the system is not optimized up to $t = 10$ min, the efficiency is low in this period. Although we start at a conservative operating point, the algorithm overestimates the adaptation of U_{cell} in the first RTO iteration. This results in a slight violation of the constraint between 10 and 20 min. Convergence is reached at the second iteration. At the end of the third RTO period, at time $t = 40$ min, the setpoint is changed to $P_{\text{el}}^{\text{S}} = 0.2 \frac{\text{W}}{\text{cm}^2}$. As a result, the fuel and air flow rates and the current are reduced, and efficiency goes up. The active constraint is now FU . This constraint is not violated since it depends only on the input variables that are not subject to plant-model mismatch. At $t = 70$ min, the setpoint is changed back to $P_{\text{el}}^{\text{S}} = 0.4 \frac{\text{W}}{\text{cm}^2}$ and, again, there is violation of the U_{cell} constraint.

Accuracy of Constraint Adaptation.

The accuracy of the constraint-adaptation scheme upon convergence is illustrated in Table 3.8. The performance loss, η_{loss} , is computed as,

$$\eta_{\text{loss}} = \frac{\eta_p^* - \eta_{\infty}^*}{\eta_p^*} \quad (3.59)$$

where η_p^* is the true optimum of the (simulated) plant, and η_{∞}^* is the objective function value obtained upon convergence of constraint adaptation. The resulting optimality loss is negligible in spite of the presence of model mismatch.

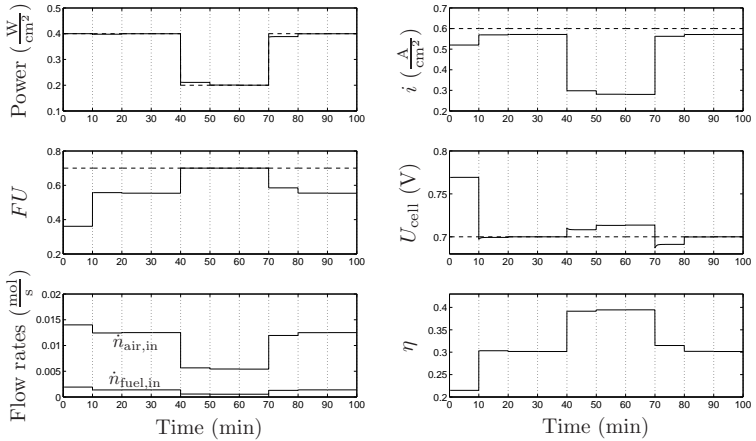


Fig. 3.16. Solid lines: Constraint-adaptation results. Dashed lines: Power setpoint and constraint bounds. The three inputs are the flow rates $\dot{n}_{\text{fuel,in}}$ and $\dot{n}_{\text{air,in}}$, and the current I , which is considered proportional to the current density i .

Table 3.8. Accuracy of the constraint-adaptation scheme.

	$P_{\text{el}}^{\text{S}} = 0.2 \frac{\text{W}}{\text{cm}^2}$		$P_{\text{el}}^{\text{S}} = 0.4 \frac{\text{W}}{\text{cm}^2}$	
	True optimum	Constraint adaptation	True optimum	Constraint adaptation
$u_1 \left(\frac{\text{mol}}{\text{s}} \right)$	5.186×10^{-4}	5.187×10^{-4}	1.338×10^{-3}	1.337×10^{-3}
$u_2 \left(\frac{\text{mol}}{\text{s}} \right)$	1.142×10^{-3}	1.138×10^{-3}	2.575×10^{-3}	2.627×10^{-3}
$u_3 \text{ (A)}$	14.012	14.013	28.571	28.571
η	0.3946	0.3946	0.3016	0.3015
η_{loss}		4.37×10^{-7}		3.69×10^{-5}

Combination of Constraint Adaptation with MPC

The same initial input and power setpoint changes as for constraint adaptation alone are applied. Constraint adaptation and control is started at $t = 10$ min with $\mathbf{B} = \mathbf{I}_2$.

Since there are three input variables and one equality constraint, no more than two (independent) inequality constraints can be active simultaneously. For this SOFC system, the bounds on T_{stack} do not become active with varying power demand. Furthermore, since there is near collinearity between U_{cell} , FU and i , these constraints are not controlled simultaneously. Hence, the quadratic objective function to be minimized by MPC can be chosen as:

$$J_k(\mathbf{u}(t)) = \sum_{l=1}^p \left\{ \rho^2 (P_{\text{el}}(t+l|t) - P_{\text{el}}^{\text{S}})^2 + \mathbf{q}_{a,k}^2 (\bar{G}_{a,k}(t+l|t) - \bar{G}_{a,k}^{\text{U}})^2 + \Delta \mathbf{u}(t+l-1)^{\text{T}} \mathbf{R}^{\text{T}} \mathbf{R} \Delta \mathbf{u}(t+l-1) + \mathbf{c}^2 \xi_k(t+l-1)^2 \right\} \quad (3.60)$$

where $\bar{G}_{a,k} = (\bar{\mathbf{G}})_{r,k}$ is a constraint that is active during the k th RTO iteration, chosen from among the constraints $U_{\text{cell}}^{\text{L}} \leq U_{\text{cell}}$, $FU \leq FU^{\text{U}}$ and $i \leq i^{\text{U}}$ in Problem (3.58). The remaining degree of freedom is fixed by selecting $\mathbf{A}_{k+1} = [u_{2,k+1}^*, -u_{1,k+1}^*, 0]^{\text{T}}$ (see (3.53)). This choice of \mathbf{A}_{k+1} is equivalent to fixing the excess air ratio λ_{air} to its optimal value given by constraint adaptation at iteration k . An equivalent option is to directly include λ_{air} as a second controlled inequality constraint, whether it is active or not at the optimum.

Combination of MPC with the constraint-adaptation scheme is illustrated schematically in Figure 3.17. At the k th RTO iteration, the optimal solution generated by the constraint-adaptation level is passed to the MPC level in the form of information regarding (i) the active set \mathcal{A}_{k+1} , which indicates the inequality constraint $\bar{G}_{a,k}$ to be controlled, and (ii) an optimal target for the additional degree of freedom, given by \mathbf{A}_{k+1} and \mathbf{u}_{k+1}^* .

A time step of 2 s is chosen for MPC. The step response model is obtained for $\mathbf{u} = [8.75 \times 10^{-4}, 71.5 \times 10^{-4}, 20.00]^{\text{T}}$ and its truncation order is $n = 50$. The length of the control and prediction horizons are $m = 6$ and $p = 9$, respectively.

The performance of MPC is highly dependent on the weights chosen for the different terms in the objective function and the bounds on the input moves. These bounds for the flow rates are chosen as $\Delta u_1^{\text{U}} = 5 \times 10^{-3}$ and $\Delta u_2^{\text{U}} = 8.33 \times 10^{-2}$. No such bound is used for the current as this would hinder quick tracking of the power setpoint. The weighting matrix for the rate of change of the inputs is $\mathbf{R} = \text{diag}(10^{-4}, 10^{-2}, 10^{-1})$.

For the other weights, two different cases are presented in Table 3.9. In Case 1, tracking of the power setpoint is favored over that of the

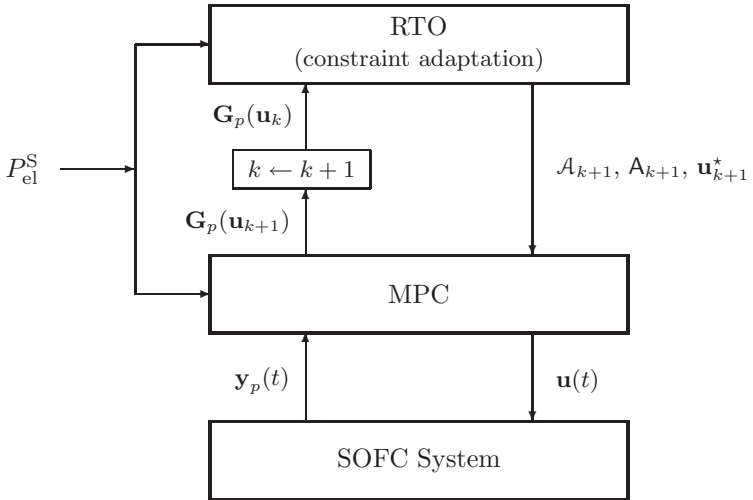


Fig. 3.17. Combination of MPC with constraint adaptation.

Table 3.9. MPC Weights

	ρ	$q_{a,k}$	c
Case 1	5	0.005	0.01
Case 2	0.001	0.05	1

active inequality constraint and the optimal value of the additional degree of freedom. The response is shown in Figure 3.18. The power tracking is virtually instantaneous, the power reaches its setpoint in about 20 s. However, this aggressive policy leads to small damped oscillations when the setpoint is changed at $t = 40$ min and $t = 70$ min. For instance, at $t = 70$ min, an abrupt increase of the air flow rate is observed, which results in a decrease of the efficiency.

A less aggressive set of weights is used in Case 2, for which the smoother response is shown in Figure 3.19. The peaks and damped oscillations are eliminated at the expense of a slower tracking of the power setpoint, which is now reached within 2-3 min of the change. Note that the constraints are respected in both cases .

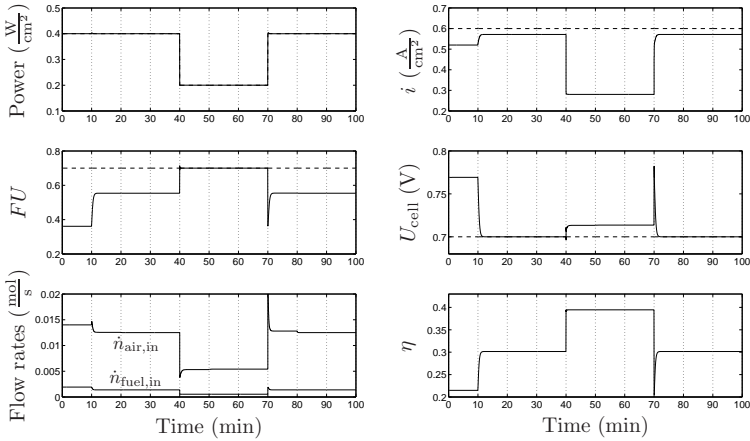


Fig. 3.18. Solid lines: Constraint adaptation with MPC for Case 1. Dashed lines: Power setpoint and constraint bounds. The three inputs are the flow rates $\dot{n}_{\text{fuel,in}}$ and $\dot{n}_{\text{air,in}}$, and the current I , which is considered proportional to the current density i .

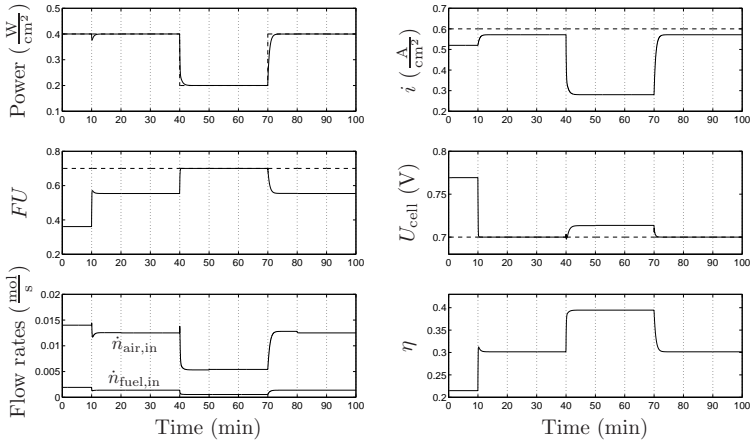


Fig. 3.19. Solid lines: Constraint adaptation with MPC for Case 2. Dashed lines: Power setpoint and constraint bounds. The three inputs are the flow rates $\dot{n}_{\text{fuel,in}}$ and $\dot{n}_{\text{air,in}}$, and the current I , which is considered proportional to the current density i .

P-i Curve.

The power density vs. current density curves are shown in Figure 3.20. The location of the optimal operating points obtained upon convergence of the RTO-MPC scheme for the two different power setpoints are clearly indicated. In both cases, optimal operation is on the left side of the maximum power density. The constraints on current density, cell potential and fuel utilization have prevented the operating point from crossing to the right of the maximum power density. Note that the step response model used by MPC was obtained on the left side of the maximum power density and thus would become inadequate if the plant operation crosses to the right side. Golbert and Lewin (2007) [44] have reported oscillatory behavior when the MPC model and the plant are on different sides of the maximum power density.

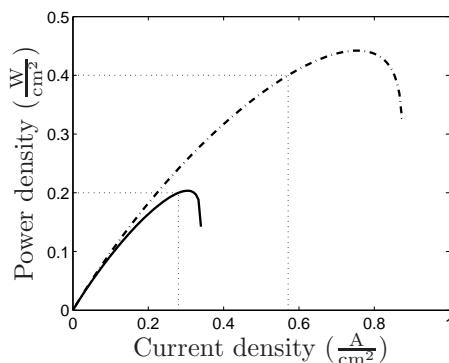


Fig. 3.20. Location of the optimum operating point upon convergence of the RTO-MPC scheme for the two different power-density setpoints of 0.2 and 0.4 $\frac{\text{W}}{\text{cm}^2}$.

3.4 Summary

In this chapter, constraint-adaptation algorithms have been considered in the context of real-time optimization. The underlying idea is to use measurements to adapt the process constraints at each RTO iteration, while relying on a fixed process model for satisfying the remaining part

of the NCO. Constraint-adaptation algorithms are based on the observation that, for many problems, most of the optimization potential arises from activating the correct set of constraints. In Section 3.1, a theoretical justification of these methods, which is based on a variational analysis of the cost function, has been provided in the case of moderate parametric uncertainty. Then, in Section 3.2, various aspects of the constraint-adaptation algorithm have been discussed and illustrated through numerical examples, including the detection of active constraints and convergence issues. The case study of an isothermal stirred-tank reactor has been presented in Subsection 3.2.8, which illustrates the applicability and suitability of this approach.

When the optimal solution lies on the constraints of the RTO problem, constraint adaptation provides fast improvement within a small number of RTO iterations. Moreover, this approach does not require that the gradient of the cost and constraint functions be estimated experimentally, which makes it less sensitive to measurement noise than many other RTO methods. However, the constraint-adaptation algorithm may terminate at a suboptimal, yet feasible, point upon convergence. Moreover, this loss of optimality directly depends on the quality of the (fixed) process model.

It should also be noted that, unlike many model-free RTO methods, constraint adaptation allows the handling of constraints without having to make any assumption regarding the active constraints at the optimal point. A major advantage with respect to two-step RTO methods is that the nominal model that is used does not require refinement, thereby avoiding the conflict between the identification and optimization objectives. These features, together with constraint satisfaction upon convergence and fast convergence, make the constraint-adaptation approach much appealing for RTO applications.

If the constrained variables can be measured (or estimated) online at a sampling period much smaller than the time constant of the controlled plant, it is possible to deal with constraint violations prior to convergence of constraint adaptation by controlling the constraints that define optimal operation. An optimization scheme combining constraint adaptation with an MPC constraint controller has been proposed in Section 3.3. This scheme presents two important features: (i) An alternative constraint-adaptation scheme has been proposed that permits to update the setpoints for the constraints in the constraint controller at each iteration. The setpoints for the active constraints reach their actual constraint bound upon convergence. And (ii), the

remaining degrees of freedom are controlled to their optimal values along sensitivity-seeking directions.

Finally, the real-time optimization of a simple SOFC system has been considered in Subsection 3.3.3 to respond to the present technological need to efficiently operate fuel cell systems while ensuring that the operation remains within a safe region. A lumped dynamic model was used for simulation purposes, which considers the electrochemical, energy and mass balances taking place inside the cell. An optimization problem has been formulated to maximize electrical efficiency at a given power density, while respecting a number of operational constraints. It is shown that the constraints that determine optimal operation vary with the power density. Furthermore, in the presence of model mismatch, the optimum given by the model may not provide a feasible operating point, and not even the correct set of active constraints. The optimization scheme combining constraint adaptation at the RTO level with MPC constraint control at the process control level has been illustrated through simulation. The approach can be applied to more elaborate fuel cell systems, where ensuring safety and operational constraints and maximizing efficiency is important. Experimental validation of the results presented in Subsection 3.3.3 is foreseen for an experimental SOFC system available at LENI in EPFL.

Real-Time Optimization via Modifier Adaptation

The constraint-adaptation scheme studied in the previous chapter guarantees feasibility upon convergence. It uses measurements of the constraints in order to modify the constraint values in the optimization problem. However, since the gradients of the constraint and cost functions are obtained from an inaccurate process model, constraint adaptation *per se* does not enforce optimality. In this chapter, experimental gradient information is incorporated into the optimization problem in order to enforce optimality. This approach, named *modifier adaptation*, is presented and analyzed in Section 4.1, including discussions on KKT matching, local convergence and model adequacy. Alternative modifier-adaptation schemes are also considered, and links to previous work are highlighted.

Approaches that have been proposed for estimating experimental gradients in the context of RTO were reviewed in Subsection 1.2.6. The dual ISOPE approach described in Subsection 2.4.4 uses the current and past operating points to estimate the gradients. In that approach, a constraint that prevents ill-conditioning in gradient computation was proposed in order to ensure sufficient information in the measurements. Following similar ideas, a constraint that enforces an upper bound on the gradient error is proposed in Section 4.2 of this thesis. This upper bound includes the effect of both truncation errors and measurement noise. This constraint is included in a dual modifier-adaptation approach.

4.1 Modifier Adaptation

4.1.1 Principles of Modifier Adaptation

The idea behind modifier adaptation is to use measurements for correcting the cost and constraint predictions between successive RTO iterations in such a way that a KKT point for the model coincides with the plant optimum [37]. In contrast to two-step RTO schemes, the model parameters $\boldsymbol{\theta}$ are not updated. Instead, a linear modification of the cost and constraint functions is implemented, which relies on so-called modifiers representing the difference between the actual plant values and the predicted values of some KKT-related quantities.

At a given operating point $\underline{\mathbf{u}}$, the modified constraint functions read:

$$\mathbf{G}_m(\mathbf{u}, \boldsymbol{\theta}) := \mathbf{G}(\mathbf{u}, \boldsymbol{\theta}) + \boldsymbol{\varepsilon} + \boldsymbol{\lambda}^{\mathbf{G}\top} (\mathbf{u} - \underline{\mathbf{u}}) \quad (4.1)$$

with the modifiers $\boldsymbol{\varepsilon} \in \mathbb{R}^{n_g}$ and $\boldsymbol{\lambda}^{\mathbf{G}} \in \mathbb{R}^{n_u \times n_g}$ given by:

$$\boldsymbol{\varepsilon} := \mathbf{G}_p(\underline{\mathbf{u}}) - \mathbf{G}(\underline{\mathbf{u}}, \boldsymbol{\theta}) \quad (4.2)$$

$$\boldsymbol{\lambda}^{\mathbf{G}\top} := \frac{\partial \mathbf{G}_p}{\partial \mathbf{u}}(\underline{\mathbf{u}}) - \frac{\partial \mathbf{G}}{\partial \mathbf{u}}(\underline{\mathbf{u}}, \boldsymbol{\theta}). \quad (4.3)$$

A graphical interpretation of the modifiers in the j th input direction for the constraint G_i is depicted in Figure 4.1. The modifier ε_i corresponds to the gap between the plant and predicted constraint values at $\underline{\mathbf{u}}$, whereas λ^{G_i} represents the difference between the slopes of $G_{p,i}$ and G_i at $\underline{\mathbf{u}}$.

Likewise, the cost function is corrected as:

$$\Phi_m(\mathbf{u}, \boldsymbol{\theta}) := \Phi(\mathbf{u}, \boldsymbol{\theta}) + \boldsymbol{\lambda}^{\Phi\top} \mathbf{u} \quad (4.4)$$

where the modifier $\boldsymbol{\lambda}^{\Phi} \in \mathbb{R}^{n_u}$ is given by:

$$\boldsymbol{\lambda}^{\Phi\top} := \frac{\partial \Phi_p}{\partial \mathbf{u}}(\underline{\mathbf{u}}) - \frac{\partial \Phi}{\partial \mathbf{u}}(\underline{\mathbf{u}}, \boldsymbol{\theta}). \quad (4.5)$$

Observe that the cost modification comprises only a linear term in \mathbf{u} , as the addition of the constant term $\left(\Phi_p(\underline{\mathbf{u}}) - \Phi(\underline{\mathbf{u}}, \boldsymbol{\theta}) - \boldsymbol{\lambda}^{\Phi\top} \underline{\mathbf{u}} \right)$ to the cost function does not affect the solution point.

The modifiers and KKT-related quantities in (4.2), (4.3) and (4.5) can be denoted collectively as n_K -dimensional vectors,

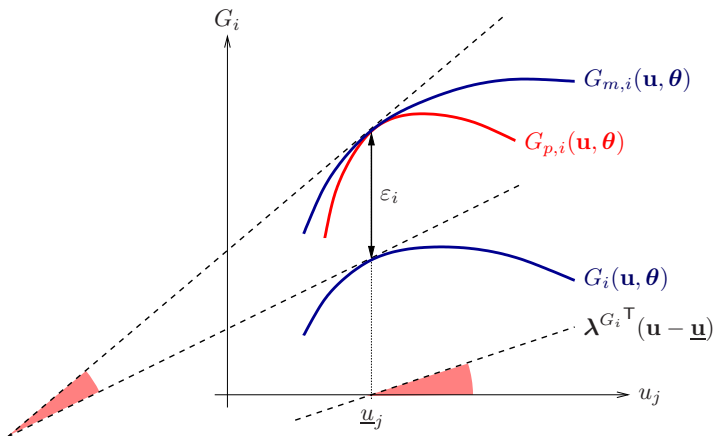


Fig. 4.1. Linear modification of the constraint function G_i so that the value and gradient of the modified function $G_{m,i}$ match those of $G_{p,i}$ at $\underline{\mathbf{u}}$.

$$\begin{aligned} \Lambda^T &:= \left(\varepsilon_1, \dots, \varepsilon_{n_g}, \lambda^{G_1^T}, \dots, \lambda^{G_{n_g}^T}, \lambda^{\Phi^T} \right) \\ \mathcal{C}^T &:= \left(G_1, \dots, G_{n_g}, \frac{\partial G_1}{\partial \mathbf{u}}, \dots, \frac{\partial G_{n_g}}{\partial \mathbf{u}}, \frac{\partial \Phi}{\partial \mathbf{u}} \right), \end{aligned}$$

with $n_K = n_g + n_u(n_g + 1)$. This way, (4.2), (4.3) and (4.5) can be rewritten as:

$$\Lambda(\underline{\mathbf{u}}) = \mathcal{C}_p(\underline{\mathbf{u}}) - \mathcal{C}(\underline{\mathbf{u}}, \theta). \quad (4.6)$$

Implementation of these modifications requires the cost and constraint gradients of the plant, $\frac{\partial \Phi_p}{\partial \mathbf{u}}(\underline{\mathbf{u}})$ and $\frac{\partial \mathbf{G}_p}{\partial \mathbf{u}}(\underline{\mathbf{u}})$, to be available at $\underline{\mathbf{u}}$. These gradients can be inferred from the measured plant outputs $\mathbf{y}_p(\underline{\mathbf{u}})$ and the estimated output gradients $\frac{\partial \mathbf{y}_p}{\partial \mathbf{u}}(\underline{\mathbf{u}})$:

$$\begin{aligned} \frac{\partial \Phi_p}{\partial \mathbf{u}}(\underline{\mathbf{u}}) &= \frac{\partial \phi}{\partial \mathbf{u}}(\underline{\mathbf{u}}, \mathbf{y}_p(\underline{\mathbf{u}})) + \frac{\partial \phi}{\partial \mathbf{y}}(\underline{\mathbf{u}}, \mathbf{y}_p(\underline{\mathbf{u}})) \frac{\partial \mathbf{y}_p}{\partial \mathbf{u}}(\underline{\mathbf{u}}) \\ \frac{\partial \mathbf{G}_p}{\partial \mathbf{u}}(\underline{\mathbf{u}}) &= \frac{\partial \mathbf{g}}{\partial \mathbf{u}}(\underline{\mathbf{u}}, \mathbf{y}_p(\underline{\mathbf{u}})) + \frac{\partial \mathbf{g}}{\partial \mathbf{y}}(\underline{\mathbf{u}}, \mathbf{y}_p(\underline{\mathbf{u}})) \frac{\partial \mathbf{y}_p}{\partial \mathbf{u}}(\underline{\mathbf{u}}). \end{aligned}$$

A discussion of how to estimate the gradients for the plant is deferred to Subsection 4.2.1.

The proposed modifier-adaptation scheme is depicted in Figure 4.2. It consists in applying the foregoing modification procedure to deter-

mine the new operating point. In the k th iteration, the next point \mathbf{u}_{k+1} is obtained as:

$$\mathbf{u}_{k+1} := \mathbf{u}_{k+1}^*, \quad (4.7)$$

where

$$\begin{aligned} \mathbf{u}_{k+1}^* \in \arg \min_{\mathbf{u}} \Phi_m(\mathbf{u}, \boldsymbol{\theta}) &= \Phi(\mathbf{u}, \boldsymbol{\theta}) + \boldsymbol{\lambda}_k^{\Phi \top} \mathbf{u} \\ \text{s.t. } \mathbf{G}_m(\mathbf{u}, \boldsymbol{\theta}) &= \mathbf{G}(\mathbf{u}, \boldsymbol{\theta}) + \boldsymbol{\varepsilon}_k + \boldsymbol{\lambda}_k^{\mathbf{G} \top} (\mathbf{u} - \mathbf{u}_k) \leq \mathbf{0} \\ \mathbf{u}^{\text{L}} &\leq \mathbf{u} \leq \mathbf{u}^{\text{U}}. \end{aligned} \quad (4.8)$$

Here, \mathbf{u}_k is the current operating point; $\boldsymbol{\varepsilon}_k$ and $\boldsymbol{\lambda}_k^{\mathbf{G}}$ are the constraint-value and constraint-gradient modifiers at the current iteration; and $\boldsymbol{\lambda}_k^{\Phi}$ is the cost-gradient modifier at the current iteration. These modifiers are adapted repeatedly by using (estimates of) the constraint values and cost and constraint gradients of the plant at \mathbf{u}_k .

The simplest adaptation strategy is to implement the full modification given by (4.6) at each iteration:

$$\boldsymbol{\Lambda}_{k+1} = \mathbf{C}_p(\mathbf{u}_{k+1}) - \mathbf{C}(\mathbf{u}_{k+1}, \boldsymbol{\theta}). \quad (4.9)$$

However, this simple strategy may lead to excessive correction when operating far away from the optimum, and it may also make modifier adaptation very sensitive to measurement noise. A better strategy consists in filtering the modifiers, e.g., with a first-order exponential filter:

$$\boldsymbol{\Lambda}_{k+1} = (\mathbf{I} - \mathbf{K})\boldsymbol{\Lambda}_k + \mathbf{K}[\mathbf{C}_p(\mathbf{u}_{k+1}) - \mathbf{C}(\mathbf{u}_{k+1}, \boldsymbol{\theta})] \quad (4.10)$$

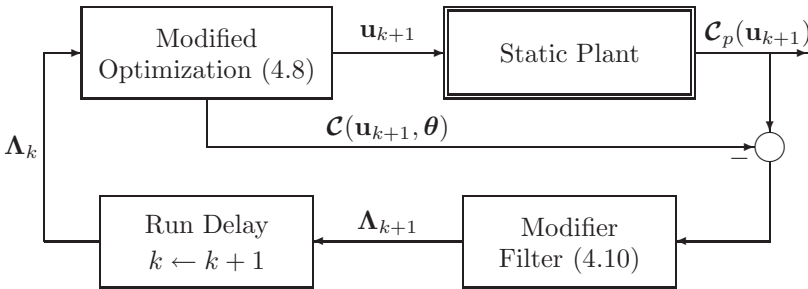


Fig. 4.2. Modifier-adaptation approach for real-time optimization.

where $\mathbf{K} \in \mathbb{R}^{n_K \times n_K}$ is a gain matrix. A possible choice for \mathbf{K} is the block-diagonal matrix

$$\mathbf{K} := \text{diag} (b_1, \dots, b_{n_g}, q_1 \mathbf{I}_{n_u}, \dots, q_{n_g} \mathbf{I}_{n_u}, d \mathbf{I}_{n_u}),$$

where the gain entries $b_1, \dots, b_{n_g}, q_1, \dots, q_{n_g}, d$ are taken in $(0, 1]$. A block-diagonal gain matrix has the advantage that it naturally decouples the modifier adaptation laws as:

$$\varepsilon_{i,k+1} = (1 - b_i) \varepsilon_{i,k} + b_i [G_{p,i}(\mathbf{u}_{k+1}) - G_i(\mathbf{u}_{k+1}, \boldsymbol{\theta})], \quad (4.11)$$

$$i = 1, \dots, n_g$$

$$\boldsymbol{\lambda}_{k+1}^{G_i \top} = (1 - q_i) \boldsymbol{\lambda}_k^{G_i \top} + q_i \left[\frac{\partial G_{p,i}}{\partial \mathbf{u}}(\mathbf{u}_{k+1}) - \frac{\partial G_i}{\partial \mathbf{u}}(\mathbf{u}_{k+1}, \boldsymbol{\theta}) \right], \quad (4.12)$$

$$i = 1, \dots, n_g$$

$$\boldsymbol{\lambda}_{k+1}^{\Phi \top} = (1 - d) \boldsymbol{\lambda}_k^{\Phi \top} + d \left[\frac{\partial \Phi_p}{\partial \mathbf{u}}(\mathbf{u}_{k+1}) - \frac{\partial \Phi}{\partial \mathbf{u}}(\mathbf{u}_{k+1}, \boldsymbol{\theta}) \right]. \quad (4.13)$$

More general choices of the gain matrix are of course possible but are typically more difficult to make. The condition for local convergence that will be introduced in Subsection 4.1.4 can be used as a basis for providing guidelines for choosing the elements of the gain matrix.

It may happen that the constraint and cost gradients cannot be reliably estimated due to the particular process characteristics or high noise level (see Subsection 4.2.1). In this case, one may decide not to adapt the gradient modifiers, e.g., by setting $q_1 = \dots = q_{n_g} = d = 0$; that is, the modifier-adaptation algorithm reduces to the constraint-adaptation scheme of Chapter 3.

The computational complexity of the modifier-adaptation algorithm is dictated by the solution of the NLP subproblems. The complexity of the NLP subproblems is similar to that of the optimization step in the classical two-step approach. Modifier adaptation is actually less computationally demanding than the classical two-step approach in that the solution of a parameter estimation problem is no longer needed at each iteration.

4.1.2 Modifier-Adaptation Algorithm

Formal statement and analysis of the modifier-adaptation algorithm requires a number of definitions. An auxiliary constraint modifier $\hat{\varepsilon}$

is introduced first, which includes the constant terms in the modified constraints (4.1),

$$\hat{\boldsymbol{\varepsilon}} := \boldsymbol{\varepsilon} - \boldsymbol{\lambda}^{\mathbf{G}\top} \mathbf{u}.$$

The modified optimization problem (4.8) expressed in terms of the auxiliary modifiers $\hat{\boldsymbol{\varepsilon}}$ reads:

$$\begin{aligned} \mathbf{u}_{k+1}^* \in \arg \min_{\mathbf{u}} \Phi_m(\mathbf{u}, \boldsymbol{\theta}) &= \Phi(\mathbf{u}, \boldsymbol{\theta}) + \boldsymbol{\lambda}_k^{\Phi\top} \mathbf{u} & (4.14) \\ \text{s.t. } \mathbf{G}_m(\mathbf{u}, \boldsymbol{\theta}) &= \mathbf{G}(\mathbf{u}, \boldsymbol{\theta}) + \hat{\boldsymbol{\varepsilon}}_k + \boldsymbol{\lambda}_k^{\mathbf{G}\top} \mathbf{u} \leq \mathbf{0} \\ \mathbf{u}^L &\leq \mathbf{u} \leq \mathbf{u}^U. \end{aligned}$$

The corresponding n_K -dimensional vector of modifiers is denoted by

$$\hat{\boldsymbol{\Lambda}}^\top := \left(\hat{\boldsymbol{\varepsilon}}_1, \dots, \hat{\boldsymbol{\varepsilon}}_{n_g}, \boldsymbol{\lambda}^{G_1\top}, \dots, \boldsymbol{\lambda}^{G_{n_g}\top}, \boldsymbol{\lambda}^{\Phi\top} \right),$$

and is related to $\boldsymbol{\Lambda}$ as:

$$\boldsymbol{\Lambda}_k = \mathbb{T}(\mathbf{u}_k) \hat{\boldsymbol{\Lambda}}_k,$$

with the matrix $\mathbb{T}(\mathbf{u}) \in \mathbb{R}^{n_K \times n_K}$ given by

$$\mathbb{T}(\mathbf{u}) := \begin{pmatrix} 1 & & & & & & & & & & \mathbf{u}^\top \\ & \ddots & & & & & & & & & \\ & & 1 & & & & & & & & \mathbf{u}^\top \\ & & & \mathbf{I}_{n_u} & & & & & & & \\ & & & & \ddots & & & & & & \\ & & & & & \mathbf{I}_{n_u} & & & & & \\ & & & & & & \mathbf{I}_{n_u} & & & & \\ & & & & & & & \mathbf{I}_{n_u} & & & \end{pmatrix}.$$

The *feasible solution map*, U , is defined as

$$U(\hat{\boldsymbol{\Lambda}}) := \{\mathbf{u} \in [\mathbf{u}^L, \mathbf{u}^U] : \mathbf{G}(\mathbf{u}, \boldsymbol{\theta}) + \hat{\boldsymbol{\varepsilon}} + \boldsymbol{\lambda}_k^{\mathbf{G}\top} \mathbf{u} \leq \mathbf{0}\},$$

such that $U(\hat{\boldsymbol{\Lambda}})$ is a subset of \mathbb{R}^{n_u} for each $\hat{\boldsymbol{\Lambda}} \in \mathbb{R}^{n_K}$. Observe that $U(\hat{\boldsymbol{\Lambda}})$ may be empty for certain values of $\hat{\boldsymbol{\Lambda}}$, which motivates the definition of the *domain of U* as

$$\text{dom } U := \{\hat{\boldsymbol{\Lambda}} \in \mathbb{R}^{n_K} : U(\hat{\boldsymbol{\Lambda}}) \neq \emptyset\}.$$

The *optimal value function*, Φ_m^* , and the *optimal solution map*, U^* , are defined as

$$\Phi_m^*(\hat{\Lambda}) := \begin{cases} \inf\{\Phi(\mathbf{u}, \boldsymbol{\theta}) + \boldsymbol{\lambda}_k^{\Phi^\top} \mathbf{u} : \mathbf{u} \in U(\hat{\Lambda})\}, & \text{if } \hat{\Lambda} \in \text{dom } U \\ +\infty, & \text{otherwise} \end{cases}$$

$$U^*(\hat{\Lambda}) := \{\mathbf{u} \in U(\hat{\Lambda}) : \Phi(\mathbf{u}, \boldsymbol{\theta}) + \boldsymbol{\lambda}_k^{\Phi^\top} \mathbf{u} = \Phi_m^*(\hat{\Lambda})\}.$$

The cost function of the modified problem being continuous and its feasible region being bounded for $\hat{\Lambda}$ and $\boldsymbol{\theta}$ given, it follows that the infimum $\Phi_m^*(\hat{\Lambda})$ is assumed somewhere on $[\mathbf{u}^L, \mathbf{u}^U]$ for each $\hat{\Lambda} \in \text{dom } U$ by Weierstrass' theorem (see, e.g., [7], Theorem 2.3.1); $\Phi_m^*(\hat{\Lambda})$ is thus a minimum. Besides the existence of a minimum point, we make the following uniqueness assumption throughout:

Assumption 4.1 *For each $\hat{\Lambda} \in \text{dom } U$, the modified optimization problem (4.8) has a unique (global) solution point.*

This assumption implies that $U^*(\hat{\Lambda})$ is a singleton, for each $\hat{\Lambda} \in \text{dom } U$. That is, at any RTO iteration k we have $\mathbf{u}_{k+1}^* := U^*(\hat{\Lambda}_k)$, if $\hat{\Lambda}_k \in \text{dom } U$. This uniqueness assumption precludes multiple global solution points, but it assumes nothing regarding the existence of local optima. Uniqueness of the global optimum is required in order to establish convergence. If the model-based optimization problem exhibited several global solutions, the optimizer would randomly pick either one of these global solution points at any iteration, thereby making convergence hopeless.

Consider the map $\Gamma(\hat{\Lambda})$ that represents the difference between the plant and predicted KKT quantities \mathcal{C} for a given set of auxiliary modifiers $\hat{\Lambda}$:

$$\Gamma(\hat{\Lambda}) := \mathcal{C}_p(\mathbf{u}) - \mathcal{C}(\mathbf{u}, \boldsymbol{\theta}), \quad \mathbf{u} = U^*(\hat{\Lambda}). \quad (4.15)$$

With this, the modifier-update law (4.10) can be rewritten as:

$$\mathbb{T}(\mathbf{u}_{k+1}) \hat{\Lambda}_{k+1} = (\mathbf{I} - \mathbf{K}) \mathbb{T}(\mathbf{u}_k) \hat{\Lambda}_k + \mathbf{K} \Gamma(\hat{\Lambda}_k).$$

Noting that \mathbb{T} is invertible, with $(\mathbb{T}(\mathbf{u}))^{-1} = \mathbb{T}(-\mathbf{u})$, this latter law can then be written in the generic form:

$$\hat{\Lambda}_{k+1} = \mathcal{M}(\hat{\Lambda}_k, \hat{\Lambda}_{k-1}), \quad (4.16)$$

with

$$\begin{aligned} \mathcal{M}(\hat{\Lambda}_k, \hat{\Lambda}_{k-1}) := & \mathsf{T}(-U^*(\hat{\Lambda}_k)) (\mathbf{I} - \mathsf{K}) \mathsf{T}(U^*(\hat{\Lambda}_{k-1})) \hat{\Lambda}_k \\ & + \mathsf{T}(-U^*(\hat{\Lambda}_k)) \mathsf{K} \mathbf{\Gamma}(\hat{\Lambda}_k). \end{aligned} \quad (4.17)$$

4.1.3 KKT Matching

Perhaps the most attractive property of modifier-adaptation schemes is that, upon convergence (under noise-free conditions), the resulting KKT point \mathbf{u}_∞ for the modified model-based optimization problem (4.8) is also a KKT point for the plant optimization problem (2.2). This is formalized in the following theorem.

Theorem 4.1 (KKT matching) *Let the gain matrix K be nonsingular and assume that the modifier-adaptation algorithm $\mathcal{M}(\hat{\Lambda}_k, \hat{\Lambda}_{k-1})$ converges, with $\mathbf{u}_\infty := \lim_{k \rightarrow \infty} \mathbf{u}_k$ being a KKT point for the modified problem (4.8). Then, \mathbf{u}_∞ is also a KKT point for the plant.*

Proof. Since K is nonsingular, letting $k \rightarrow \infty$ in (4.10) gives:

$$\Lambda_\infty = \mathbf{C}_p(\mathbf{u}_\infty) - \mathbf{C}(\mathbf{u}_\infty, \boldsymbol{\theta}), \quad (4.18)$$

that is,

$$\varepsilon_{i,\infty} = G_{p,i}(\mathbf{u}_\infty) - G_i(\mathbf{u}_\infty, \boldsymbol{\theta}), \quad i = 1, \dots, n_g \quad (4.19)$$

$$\boldsymbol{\lambda}_\infty^{\mathbf{G}_i \mathsf{T}} = \frac{\partial G_{p,i}}{\partial \mathbf{u}}(\mathbf{u}_\infty) - \frac{\partial G_i}{\partial \mathbf{u}}(\mathbf{u}_\infty, \boldsymbol{\theta}), \quad i = 1, \dots, n_g \quad (4.20)$$

$$\boldsymbol{\lambda}_\infty^{\boldsymbol{\Phi} \mathsf{T}} = \frac{\partial \Phi_p}{\partial \mathbf{u}}(\mathbf{u}_\infty) - \frac{\partial \Phi}{\partial \mathbf{u}}(\mathbf{u}_\infty, \boldsymbol{\theta}). \quad (4.21)$$

It is then readily seen that, upon convergence, the KKT-elements \mathbf{C}_m for the modified problem (4.8) match the corresponding elements \mathbf{C}_p for the plant:

$$\mathbf{C}_m(\mathbf{u}_\infty, \boldsymbol{\theta}) = \mathbf{C}(\mathbf{u}_\infty, \boldsymbol{\theta}) + \Lambda_\infty = \mathbf{C}_p(\mathbf{u}_\infty), \quad (4.22)$$

or, considering the individual terms:

$$\mathbf{G}_m(\mathbf{u}_\infty, \boldsymbol{\theta}) = \mathbf{G}(\mathbf{u}_\infty, \boldsymbol{\theta}) + \varepsilon_\infty = \mathbf{G}_p(\mathbf{u}_\infty) \quad (4.23)$$

$$\frac{\partial \mathbf{G}_m}{\partial \mathbf{u}}(\mathbf{u}_\infty, \boldsymbol{\theta}) = \frac{\partial \mathbf{G}}{\partial \mathbf{u}}(\mathbf{u}_\infty, \boldsymbol{\theta}) + \boldsymbol{\lambda}_\infty^{\mathbf{G} \mathsf{T}} = \frac{\partial \mathbf{G}_p}{\partial \mathbf{u}}(\mathbf{u}_\infty) \quad (4.24)$$

$$\frac{\partial \Phi_m}{\partial \mathbf{u}}(\mathbf{u}_\infty, \boldsymbol{\theta}) = \frac{\partial \Phi}{\partial \mathbf{u}}(\mathbf{u}_\infty, \boldsymbol{\theta}) + \boldsymbol{\lambda}_\infty^{\boldsymbol{\Phi} \mathsf{T}} = \frac{\partial \Phi_p}{\partial \mathbf{u}}(\mathbf{u}_\infty). \quad (4.25)$$

Since, by assumption, \mathbf{u}_∞ is a KKT point for the modified problem (4.8), it satisfies equations (2.5)-(2.8) with the associated Lagrange multipliers $\boldsymbol{\mu}_\infty$, $\boldsymbol{\zeta}_\infty^U$ and $\boldsymbol{\zeta}_\infty^L$. Hence, from (4.23)-(4.25), \mathbf{u}_∞ is also a KKT point for the original problem (2.2), with the same set of Lagrange multipliers. \square

A direct consequence of Theorem 4.1 is that, at \mathbf{u}_∞ , the active constraints and the corresponding Lagrange multipliers are the same for the modified problem (4.8) and the plant problem (2.2). Furthermore, note that the equalities (4.23)-(4.25) represent more than is strictly needed for KKT matching: indeed, since the Lagrange multipliers $\boldsymbol{\mu}_\infty$ corresponding to inactive constraints are zero, one simply needs to match the values and gradients of the *active* constraints. However, adaptation of the inactive constraints and their gradients is required to detect the active set, which is not known *a priori*.

Note also that no guarantee can be given that a global optimizer for the plant has been determined, even if the successive operating points \mathbf{u}_{k+1} correspond to *global* solutions of the modified problem (4.8). Indeed, the converged operating point \mathbf{u}_∞ may correspond to any stationary point for the plant, e.g., also to a local minimum. A special case in which modifier-adaptation guarantees a global optimum for the plant is that of the optimization problem (2.2) being convex, although this condition can never be verified in practice.

The following numerical example illustrates the convergence of modifier adaptation to a KKT point in the convex case.

Example 4.1 Consider the following convex optimization problem:

$$\begin{aligned} \min_{\mathbf{u} \geq \mathbf{0}} \quad & \Phi(\mathbf{u}, \boldsymbol{\theta}) := (u_1 - \theta_1)^2 + 4(u_2 - 2.5)^2 & (4.26) \\ \text{s.t.} \quad & G := (u_1 - \theta_2)^2 - 4\theta_3(u_2 - \theta_4) - 2 \leq 0, \end{aligned}$$

comprising two decision variables $\mathbf{u} = [u_1 \ u_2]^\top$, four model parameters $\boldsymbol{\theta} = [\theta_1 \ \theta_2 \ \theta_3 \ \theta_4]^\top$, and a single inequality constraint G . The parameter values $\boldsymbol{\theta}$ for the plant (simulated reality) and the model are reported in Table 4.1.

Figure 4.3 illustrates the convergence of several implementations of the modifier-adaptation scheme. Note that, due to parametric errors, the plant optimum (point P) and model optimum (point M) are quite different. Starting from the model optimum, constraint adaptation alone is first applied, i.e., with $d = 0$, and $\mathbf{Q} = \mathbf{q} = \mathbf{0}$; the iterations

Table 4.1. Values of the uncertain parameters θ in Problem (4.26) corresponding to the plant and the model.

	θ_1	θ_2	θ_3	θ_4
Plant	3.5	2.5	-0.4	1.0
Model	2.0	1.5	-0.5	0.5

a with $B = b = 0.8$ converge to the feasible, yet suboptimal operating point *C*. By enabling the correction of the gradient of the cost function ($b = d = 0.8$; $q = 0$), one obtains iterations *b*, while enabling the correction of the gradient of the constraint ($b = q = 0.8$; $d = 0$) yields the iterations *c*. These two intermediate cases show how the plant optimum can be approached with respect to case *a* by correcting the different gradients involved in the KKT conditions. Finally, the full modifier-adaptation algorithm applied with $b = d = q = 0.8$ produces iterations *d*, which converge to the plant optimum.

For some problems, the iterations may converge by following an infeasible path (i.e., with violation of the constraints), even if the modifier-adaptation algorithm starts at a feasible point. A way of reducing the magnitude of a constraint violation is by reducing the gain coefficients in the matrix K ; however, this is at the expense of a slower convergence rate. Constraint violation can also be prevented by devising a way to combine modifier-adaptation with a constraint controller, as is done in Subsection 3.3 for the case of constraint adaptation alone. Even though the iterations may follow an infeasible path, a straightforward consequence of Theorem 4.1 is that a convergent modifier-adaptation scheme yields feasible operation after a *finite* number of RTO iterations upon *backing-off* the constraints in the original problem.

Theorem 4.1 establishes that, under mild conditions, a convergent implementation of the scheme (4.7), (4.8) and (4.10) finds a KKT point for the plant optimization problem (2.2). Yet, convergence is not guaranteed. It may happen, for instance, that the modified NLP problem (4.8) becomes infeasible because a modifier is too large, or that the modifier sequence exhibits undamped oscillations when some gain coefficients in the matrix K are too large. Some guidance regarding the choice of the gain matrix K is given subsequently, based on a local convergence analysis.

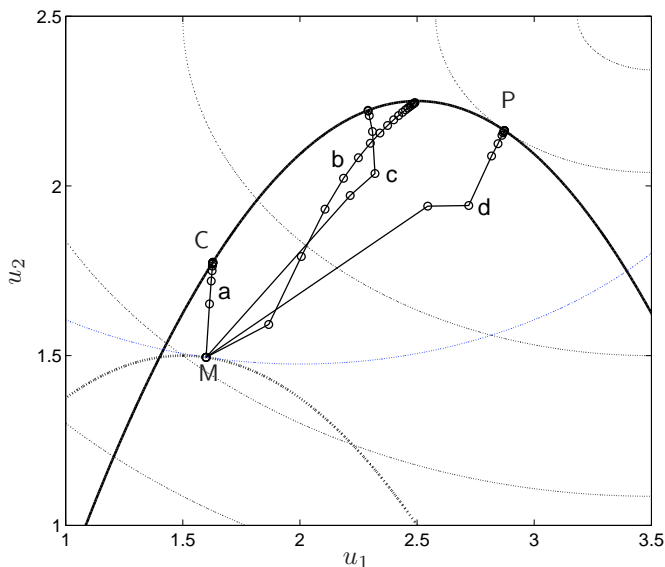


Fig. 4.3. Modifier adaptation applied to Problem (4.26). Thick solid line: constraint for the plant; Thick dash-dotted line: constraint predicted by the model; Dotted lines: contours of the cost function; Point P: plant optimum; Point M: model optimum; Point C: optimum found using only constraint adaptation.

4.1.4 Convergence Analysis

This subsection derives necessary conditions for the modifier-adaptation algorithm (4.7), (4.8) and (4.10) to converge (under noise-free conditions). To conduct the analysis, the auxiliary constraint modifier $\hat{\epsilon}$ introduced in Subsection 4.1.2, which corresponds to the sum of the constant terms in the constraint modification (4.1), is employed.

Theorem 4.2 (Asymptotic Stability - Filters on the Modifiers) *Let $\hat{\Lambda}_\infty$ be a fixed point of the modifier-adaptation algorithm \mathcal{M} , and assume that:*

1. *the second-order sufficient conditions for a local minimum of the modified problem (4.8) hold at $\mathbf{u}_\infty = U^*(\hat{\Lambda}_\infty)$, with the associated Lagrange multipliers $\boldsymbol{\mu}_\infty$, ζ_∞^U and ζ_∞^L ;*

2. the regularity condition (2.4) holds at \mathbf{u}_∞ ;
3. $\mu_{i,\infty} > 0$, $\zeta_{i,\infty}^U > 0$, and $\zeta_{i,\infty}^L > 0$ for each active constraint.

Then, a necessary condition for the local asymptotic convergence of \mathcal{M} with filter on the modifiers to the fixed point $\hat{\Lambda}_\infty$ is that the gain matrix \mathbf{K} be chosen such that

$$\varrho \left\{ \begin{pmatrix} \mathbf{A}_\infty & \mathbf{B}_\infty \\ \mathbf{I}_{n_K} & \mathbf{0}_{n_K \times n_K} \end{pmatrix} \right\} < 1, \quad (4.27)$$

with

$$\mathbf{A}_\infty = \mathbf{T}_\infty^{-1} \left(\left[\begin{pmatrix} \lambda_\infty^{\mathbf{G}^\top} \\ \mathbf{0} \end{pmatrix} - \mathbf{K} \mathbf{\Xi}_\infty \right] \mathcal{P}_u \mathbf{M}_\infty^{-1} \mathbf{N}_\infty + (\mathbf{I} - \mathbf{K}) \mathbf{T}_\infty \right), \quad (4.28)$$

$$\mathbf{B}_\infty = -\mathbf{T}_\infty^{-1} (\mathbf{I} - \mathbf{K}) \begin{pmatrix} \lambda_\infty^{\mathbf{G}^\top} \\ \mathbf{0} \end{pmatrix} \mathcal{P}_u \mathbf{M}_\infty^{-1} \mathbf{N}_\infty, \quad (4.29)$$

where $\mathbf{T}_\infty = \mathbf{T}(\mathbf{u}_\infty)$. The matrices $\mathbf{M}_\infty \in \mathbb{R}^{(n_g+3n_u) \times (n_g+3n_u)}$ and $\mathcal{P}_u \in \mathbb{R}^{n_u \times (n_g+3n_u)}$ are the same as in Theorem 3.4, and the matrices $\mathbf{N}_\infty \in \mathbb{R}^{(n_g+3n_u) \times n_K}$ and $\mathbf{\Xi}_\infty \in \mathbb{R}^{n_K \times n_u}$ are defined as

$$\mathbf{N}_\infty := \begin{pmatrix} \mathbf{0}_{n_u \times n_g} & \mu_{1,\infty} \mathbf{I}_{n_u} & \dots & \mu_{n_g,\infty} \mathbf{I}_{n_u} & \mathbf{I}_{n_u} \\ \text{diag}(\boldsymbol{\mu}_\infty) & & & & \\ \mathbf{0}_{2n_u \times n_g} & & & & \end{pmatrix},$$

$$\mathbf{\Xi}_\infty := \frac{\partial \mathcal{C}_p}{\partial \mathbf{u}}(\mathbf{u}_\infty) - \frac{\partial \mathcal{C}}{\partial \mathbf{u}}(\mathbf{u}_\infty, \boldsymbol{\theta}) = \begin{pmatrix} \lambda_\infty^{\mathbf{G}^\top} \\ \frac{\partial^2 G_{p,1}}{\partial \mathbf{u}^2} - \frac{\partial^2 G_1}{\partial \mathbf{u}^2} \\ \vdots \\ \frac{\partial^2 G_{p,n_g}}{\partial \mathbf{u}^2} - \frac{\partial^2 G_{n_g}}{\partial \mathbf{u}^2} \\ \frac{\partial^2 \Phi_p}{\partial \mathbf{u}^2} - \frac{\partial^2 \Phi}{\partial \mathbf{u}^2} \end{pmatrix}_{(\mathbf{u}_\infty)},$$

and $\varrho\{\cdot\}$ stands for the spectral radius. Moreover, if the modifier-adaptation algorithm converges, then the rate of convergence is linear.

Proof. It follows from the assumptions and Theorem 2.3 that there is some $\eta > 0$ such that, for each $\hat{\Lambda} \in \mathcal{B}_\eta(\hat{\Lambda}_\infty)$, there exists a continuously differentiable vector function $\mathbf{u}^* = U^*(\hat{\Lambda})$ satisfying the second-order sufficient conditions for a local minimum of the modified problem (4.8). Moreover, from Theorem 2.3, we have:

$$\frac{\partial U^*}{\partial \hat{\Lambda}}(\hat{\Lambda}_\infty) = -\mathcal{P}_u M_\infty^{-1} N_\infty$$

The constraint functions G_i and $G_{p,i}$, $i = 1, \dots, n_g$, being differentiable with respect to \mathbf{u} , it follows that a first order approximation of (4.10) in the neighborhood of \mathbf{u}_∞ is given by

$$\delta \mathbf{\Lambda}_{k+1} = (\mathbf{I} - \mathbf{K})\delta \mathbf{\Lambda}_k + \mathbf{K} \Xi_\infty \delta \mathbf{u}_{k+1}^* \quad (4.30)$$

where $\delta \mathbf{\Lambda}_k = \mathbf{\Lambda}_k - \mathbf{\Lambda}_\infty$ and $\delta \mathbf{u}_{k+1}^* = \mathbf{u}_{k+1}^* - \mathbf{u}_\infty$.

Replacing $\delta \mathbf{u}_{k+1}^* = -\mathcal{P}_u M_\infty^{-1} N_\infty \delta \hat{\Lambda}_k$ in (4.30) we obtain

$$\delta \mathbf{\Lambda}_{k+1} = (\mathbf{I} - \mathbf{K})\delta \mathbf{\Lambda}_k - \mathbf{K} \Xi_\infty \mathcal{P}_u M_\infty^{-1} N_\infty \delta \hat{\Lambda}_k. \quad (4.31)$$

Also, a first order approximation of $\mathbf{\Lambda}_k = \mathbb{T}(\mathbf{u}_k^*) \hat{\Lambda}_k$ in the neighborhood of \mathbf{u}_∞ gives

$$\begin{aligned} \delta \mathbf{\Lambda}_k &= \mathbb{T}_\infty \delta \hat{\Lambda}_k + \begin{pmatrix} \lambda_\infty^{\mathbf{G}^\top} \\ \mathbf{0} \end{pmatrix} \delta \mathbf{u}_k^* \\ &= \mathbb{T}_\infty \delta \hat{\Lambda}_k - \begin{pmatrix} \lambda_\infty^{\mathbf{G}^\top} \\ \mathbf{0} \end{pmatrix} \mathcal{P}_u M_\infty^{-1} N_\infty \delta \hat{\Lambda}_{k-1} \end{aligned} \quad (4.32)$$

Replacing (4.32) in (4.31) we have

$$\begin{aligned} \mathbb{T}_\infty \delta \hat{\Lambda}_{k+1} - \begin{pmatrix} \lambda_\infty^{\mathbf{G}^\top} \\ \mathbf{0} \end{pmatrix} \mathcal{P}_u M_\infty^{-1} N_\infty \delta \hat{\Lambda}_k &= (\mathbf{I} - \mathbf{K})\mathbb{T}_\infty \delta \hat{\Lambda}_k \\ - (\mathbf{I} - \mathbf{K}) \begin{pmatrix} \lambda_\infty^{\mathbf{G}^\top} \\ \mathbf{0} \end{pmatrix} \mathcal{P}_u M_\infty^{-1} N_\infty \delta \hat{\Lambda}_{k-1} - \mathbf{K} \Xi_\infty \mathcal{P}_u M_\infty^{-1} N_\infty \delta \hat{\Lambda}_k & \end{aligned}$$

and rearranging we have

$$\begin{aligned} \mathbb{T}_\infty \delta \hat{\Lambda}_{k+1} &= \left(\left[\begin{pmatrix} \lambda_\infty^{\mathbf{G}^\top} \\ \mathbf{0} \end{pmatrix} - \mathbf{K} \Xi_\infty \right] \mathcal{P}_u M_\infty^{-1} N_\infty + (\mathbf{I} - \mathbf{K})\mathbb{T}_\infty \right) \delta \hat{\Lambda}_k \\ &\quad - (\mathbf{I} - \mathbf{K}) \begin{pmatrix} \lambda_\infty^{\mathbf{G}^\top} \\ \mathbf{0} \end{pmatrix} \mathcal{P}_u M_\infty^{-1} N_\infty \delta \hat{\Lambda}_{k-1} \end{aligned}$$

which gives the linearization form of (4.16):

$$\delta \hat{\Lambda}_{k+1} = \mathbf{A}_\infty \delta \hat{\Lambda}_k + \mathbf{B}_\infty \delta \hat{\Lambda}_{k-1} \quad (4.33)$$

which can be written in the following way

$$\begin{pmatrix} \delta \hat{\Lambda}_{k+1} \\ \delta \hat{\Lambda}_k \end{pmatrix} = \Upsilon_\infty \begin{pmatrix} \delta \hat{\Lambda}_k \\ \delta \hat{\Lambda}_{k-1} \end{pmatrix},$$

with

$$\Upsilon_\infty := \begin{pmatrix} \mathbf{A}_\infty & \mathbf{B}_\infty \\ \mathbf{I}_{n_K} & \mathbf{0}_{n_K \times n_K} \end{pmatrix}. \quad (4.34)$$

Clearly, a necessary condition for the modifier-adaptation algorithm to converge to $\hat{\Lambda}_\infty$ is that the gain matrix \mathbf{K} be chosen such that the $(2n_K \times 2n_K)$ matrix Υ_∞ has spectral radius less than 1 (or any norm of Υ_∞ less than 1). Moreover, the relation (4.33) establishes the linear rate of convergence of the modifier-adaptation scheme. \square

Remark 4.1 *In the special case that $\mathbf{K} = \mathbf{I}$, condition (4.27) reduces to:*

$$\varrho \left\{ \Upsilon_\infty^{-1} \left[\begin{pmatrix} \lambda_\infty^{\mathbf{G}^\top} \\ \mathbf{0} \end{pmatrix} - \Xi_\infty \right] \mathcal{P}_u \mathbf{M}_\infty^{-1} \mathbf{N}_\infty \right\} < 1. \quad (4.35)$$

The necessary condition for convergence given in Theorem 4.2 is illustrated next.

Example 4.2 *Consider the case \mathbf{d} of Example 4.1, where modifier-adaptation algorithm is applied to Problem (4.26) with $b = d = q = 0.8$. The converged inputs and modifiers are given by:*

$$\mathbf{u}_\infty = [2.8726 \ 2.1632]^\top, \quad \hat{\Lambda}_\infty = [3.4 \ -2 \ -0.4 \ -3 \ 0]^\top.$$

At that particular point, the matrix Υ_∞ is calculated and its eigenvalues are found to be 0.427, -0.395 , 0.2 (multiplicity 4) and 0 (multiplicity 4). The spectral radius of Υ_∞ is therefore less than 1, thereby supporting the convergence shown earlier in Figure 4.3.

The next example shows how the necessary condition for convergence can be used to guide the choice of the gain matrix \mathbf{K} .

Example 4.3 *Consider the optimization problem*

$$\min_{-5 \leq u \leq 5} \Phi_p(u) := (u - 1)^2, \quad (4.36)$$

the unique solution of which is $u_p^* = 1$. Let the model of the cost function be $\Phi(u) := \theta u^2$, where $\theta = \frac{1}{4}$ is a model parameter, and consider the modified optimization problem:

$$\min_{-5 \leq u \leq 5} \theta u^2 + \lambda_k^\Phi u, \quad (4.37)$$

where the cost gradient modifier λ^Φ is adapted according to (4.13):

$$\lambda_{k+1}^\Phi = (1-d)\lambda_k^\Phi + d[2(u_{k+1}^* - 1) - 2\theta u_{k+1}^*]. \quad (4.38)$$

An unconstrained solution to Problem (4.37) is $u_{k+1}^* = -\frac{\lambda_k^\Phi}{2\theta}$, provided that $-5 \leq -\frac{\lambda_k^\Phi}{2\theta} \leq 5$. Using this solution in (4.38) gives:

$$\lambda_{k+1}^\Phi = \left(1 - \frac{d}{\theta}\right) \lambda_k^\Phi - 2d, \quad (4.39)$$

a fixed point of which is $\lambda_\infty^\Phi = -2\theta = -\frac{1}{2}$. Since $-\frac{\lambda_k^\Phi}{2\theta} \in [-5, 5]$, the assumption of an unconstrained solution is verified in the neighborhood of λ_∞^Φ .

Noting that $\Lambda = \hat{\Lambda} = \lambda^\Phi$ and $\frac{\partial \mathcal{M}}{\partial \Lambda_{k-1}}(\hat{\Lambda}_\infty) = 0$, the matrix Υ_∞ defined in (4.34) reads:

$$\Upsilon_\infty = \begin{pmatrix} 1 - \frac{d}{\theta} & 0 \\ 1 & 0 \end{pmatrix}.$$

A necessary condition for the modifier adaptation algorithm to converge is therefore $-1 < (1 - \frac{d}{\theta}) < 1$, i.e., $0 < d < \frac{1}{2}$.

4.1.5 Model Adequacy for Modifier-Adaptation Schemes

The model-adequacy requirements for two-step RTO approaches have been reviewed in Subsection 2.3.2, where they were shown to be very restrictive. In this subsection, model adequacy is investigated in the context of modifier adaptation.

A process model is called *adequate* if modifier values, say $\bar{\Lambda}$, can be found such that a fixed point of the modifier-adaptation scheme coincides with the plant optimum \mathbf{u}_p^* . The situation is much simpler than for two-step approaches for two main reasons. Firstly, by choosing $\bar{\Lambda} = \Lambda_\infty$ given by (4.22), the analogues of Conditions (2.25)-(2.27) in

Criterion 2.1 are automatically satisfied. Secondly, because the adaptation step (4.10) does not require the solution of an optimization problem, no such conditions as (2.29) and (2.30) are needed. The only remaining (*sufficient*) condition for model adequacy in modifier adaptation is therefore the analogue of (2.28). In other words, modifier values such that \mathbf{u}_p^* is a solution of the modified optimization problem (4.8) are guaranteed to exist provided that the reduced Hessian of the cost function Φ is positive definite at \mathbf{u}_p^* . Interestingly, this positive-definiteness requirement is independent of the modifier values themselves. Note that model adequacy is dictated by the second-order derivatives only, since any mismatch in the cost and constraint functions is corrected up to the first-order derivatives by the modifier-adaptation scheme. The criterion for model adequacy can be formalized as follows.

Criterion 4.1 (Model Adequacy for Modifier Adaptation) *Let \mathbf{u}_p^* be the unique plant optimum, which is assumed to be a regular point for the constraints. If the process model is such that the reduced Hessian of the cost function Φ is positive definite at \mathbf{u}_p^* ,*

$$\nabla_r^2 \Phi(\mathbf{u}_p^*, \boldsymbol{\theta}) \succ 0 \quad (\text{positive definite}), \quad (4.40)$$

then the process model is adequate for use in the modifier-adaptation RTO scheme (4.7), (4.8), (4.10).

The fact that Criterion 4.1 is much easier to meet than Criterion 2.1 represents a clear advantage of modifier-adaptation schemes over two-step approaches. In modifier adaptation, the key issue is not model adequacy, but the availability of the KKT-related quantities \mathcal{C}_p , which include plant-gradient information.

4.1.6 Alternative Schemes

This thesis argues in favor of the formulation presented in Subsection 4.1.1, where the filters are placed on the modifiers $\boldsymbol{\Lambda}$. This approach gives the ability to filter each constraint individually, i.e., it allows direct control of the constraints and helps prevent constraint violation. It also permits combination of modifier adaptation with a constraint controller as in Section 3.3.

However, a number of alternative schemes are possible and are briefly described in this subsection. These variants differ from the modifier-adaptation algorithm (4.7), (4.8) and (4.10) either in their implementation of filtering or in the way the modification itself is made.

Alternative Filtering

Instead of putting the filters on the modifiers Λ as in (4.10), one can filter the inputs \mathbf{u} . In this variant, the modifiers Λ_k are updated according to (4.6) as:

$$\Lambda_k = \mathcal{C}_p(\mathbf{u}_k) - \mathcal{C}(\mathbf{u}_k, \boldsymbol{\theta}), \quad (4.41)$$

while the next operating point is calculated by a first-order exponential filter:

$$\mathbf{u}_{k+1} = (\mathbf{I} - \mathbf{K})\mathbf{u}_k + \mathbf{K}\mathbf{u}_{k+1}^*, \quad (4.42)$$

where \mathbf{K} is a $(n_u \times n_u)$ gain matrix. Overall, this algorithm is given by (4.8),(4.41),(4.42).

Using the same notation and uniqueness assumption as in Subsection 4.1.2, a possible way of formulating the modifier-adaptation algorithm with filters on the inputs, \mathcal{M}_f , is as follows:

$$\mathbf{u}_{k+1} = \mathcal{M}_f(\mathbf{u}_k) := \mathbf{u}_k + \mathbf{K}[\mathbf{u}_{k+1}^* - \mathbf{u}_k], \quad (4.43)$$

$$\text{with: } \quad \mathbf{u}_{k+1}^* = U^*(\hat{\Lambda}_k), \quad \text{if } \hat{\Lambda}_k \in \text{dom } U, \\ \hat{\Lambda}_k = \mathbb{T}(-\mathbf{u}_k) (\mathcal{C}_p(\mathbf{u}_k) - \mathcal{C}(\mathbf{u}_k, \boldsymbol{\theta})).$$

Conditions under which this variant scheme reaches a KKT point for the plant upon convergence are the same as those stated in Theorem 4.1. This is easily seen by noting that (4.42) imposes $\mathbf{u}_\infty = \mathbf{u}_\infty^*$, and (4.18) imposes $\mathcal{C}_m(\mathbf{u}_\infty, \boldsymbol{\theta}) = \mathcal{C}_p(\mathbf{u}_\infty)$.

A local convergence analysis (under noise-free conditions) can also be performed in the neighborhood of a converged operating point \mathbf{u}_∞ .

Theorem 4.3 (Asymptotic Stability - Filters on the Inputs)

Let \mathbf{u}_∞ be a fixed point of the modifier-adaptation algorithm \mathcal{M}_f given in (4.43), and assume that:

1. the second-order sufficient conditions for a local minimum of the modified problem (4.8) hold at $\mathbf{u}_\infty = U^*(\hat{\Lambda}_\infty)$, with the associated Lagrange multipliers $\boldsymbol{\mu}_\infty$, ζ_∞^U and ζ_∞^L ;
2. the regularity condition (2.4) holds at \mathbf{u}_∞ ;
3. $\mu_{i,\infty} > 0$, $\zeta_{i,\infty}^U > 0$, and $\zeta_{i,\infty}^L > 0$ for each active constraint.

Then, a necessary condition for the local asymptotic convergence of \mathcal{M}_f to the fixed point \mathbf{u}_∞ is that the gain matrix \mathbf{K} be chosen such that

$$\varrho \left\{ \mathbf{I} - \mathbf{K} \left(\mathbf{I} + \mathcal{P}_{\mathbf{u}} \mathbf{M}_{\infty}^{-1} \mathbf{N}_{\infty} \mathbf{T}_{\infty}^{-1} \left[\boldsymbol{\Xi}_{\infty} - \begin{pmatrix} \boldsymbol{\lambda}_{\mathbf{G}}^{\mathbf{G}^{\top}} \\ \mathbf{0} \end{pmatrix} \right] \right) \right\} < 1, \quad (4.44)$$

where the matrices $\mathbf{M}_{\infty} \in \mathbf{R}^{(n_g+3n_u) \times (n_g+3n_u)}$ and $\mathcal{P}_{\mathbf{u}} \in \mathbf{R}^{n_u \times (n_g+3n_u)}$ are the same as in Theorem 3.4, the matrices $\mathbf{N}_{\infty} \in \mathbf{R}^{(n_g+3n_u) \times n_{\kappa}}$, $\boldsymbol{\Xi}_{\infty} \in \mathbf{R}^{n_{\kappa} \times n_u}$ and $\mathbf{T}_{\infty} \in \mathbf{R}^{n_{\kappa} \times n_{\kappa}}$ are the same as in Theorem 4.2, and $\varrho\{\cdot\}$ stands for the spectral radius. Moreover, if the modifier-adaptation algorithm converges, then the rate of convergence is linear.

Proof. The optimality conditions (2.8) and (2.6) for the modified optimization problem (4.8) are

$$\frac{\partial \mathcal{L}_m}{\partial \mathbf{u}}(\mathbf{u}, \boldsymbol{\theta}) = \frac{\partial \Phi_m}{\partial \mathbf{u}}(\mathbf{u}, \boldsymbol{\theta}) + \sum_{i=1}^{n_g} \mu_i \frac{\partial G_{m,i}}{\partial \mathbf{u}}(\mathbf{u}, \boldsymbol{\theta}) + \boldsymbol{\zeta}^{\mathbf{U}^{\top}} + \boldsymbol{\zeta}^{\mathbf{L}^{\top}} \quad (4.45)$$

$$\boldsymbol{\mu}^{\top} \mathbf{G}_m(\mathbf{u}, \boldsymbol{\theta}) = 0, \quad \boldsymbol{\zeta}^{\mathbf{U}^{\top}}(\mathbf{u} - \mathbf{u}^{\mathbf{U}}) = 0, \quad \boldsymbol{\zeta}^{\mathbf{L}^{\top}}(\mathbf{u}^{\mathbf{L}} - \mathbf{u}) = 0 \quad (4.46)$$

It follows from the assumptions and Theorem 2.3 that there is some $\eta > 0$ such that, for each $\hat{\boldsymbol{\Lambda}} \in \mathcal{B}_{\eta}(\hat{\boldsymbol{\Lambda}}_{\infty})$, there exists a continuously differentiable vector function $\mathbf{u}^* = U^*(\hat{\boldsymbol{\Lambda}})$ satisfying the second-order sufficient conditions for a local minimum of the modified problem (4.8). Moreover, from Theorem 2.3, we have:

$$\frac{\partial U^*}{\partial \hat{\boldsymbol{\Lambda}}}(\hat{\boldsymbol{\Lambda}}_{\infty}) = -\mathcal{P}_{\mathbf{u}} \mathbf{M}_{\infty}^{-1} \mathbf{N}_{\infty}$$

where matrix $\mathbf{M}_{\infty} \in \mathbf{R}^{(n_g+3n_u) \times (n_g+3n_u)}$ stands for the Jacobian of (4.45) and (4.46) with respect to $(\mathbf{u}, \boldsymbol{\mu}, \boldsymbol{\zeta}^{\mathbf{U}}, \boldsymbol{\zeta}^{\mathbf{L}})$ at $\hat{\boldsymbol{\Lambda}}_{\infty}$, and matrix $\mathbf{N}_{\infty} \in \mathbf{R}^{(n_g+3n_u) \times n_{\theta}}$ stands for the Jacobian of (4.45) and (4.46) with respect to $\hat{\boldsymbol{\Lambda}}$ at $\hat{\boldsymbol{\Lambda}}_{\infty}$.

Equation (4.42) can be rewritten as

$$\delta \mathbf{u}_{k+1} = (\mathbf{I} - \mathbf{K}) \delta \mathbf{u}_k + \mathbf{K} \delta \mathbf{u}_{k+1}^* \quad (4.47)$$

where $\delta \mathbf{u}_{k+1} = \mathbf{u}_{k+1} - \mathbf{u}_{\infty}$ and $\delta \mathbf{u}_{k+1}^* = \mathbf{u}_{k+1}^* - \mathbf{u}_{\infty}$. A first-order approximation of (4.47) in the neighborhood of $\hat{\boldsymbol{\Lambda}}_{\infty}$ is

$$\delta \mathbf{u}_{k+1} = (\mathbf{I} - \mathbf{K}) \delta \mathbf{u}_k - \mathbf{K} \mathcal{P}_{\mathbf{u}} \mathbf{M}_{\infty}^{-1} \mathbf{N}_{\infty} \delta \hat{\boldsymbol{\Lambda}}_k + o(\|\delta \hat{\boldsymbol{\Lambda}}_k\|) \quad (4.48)$$

Noting that the linearization form of $\boldsymbol{\Lambda}_k = \mathbf{T}(\mathbf{u}_k) \hat{\boldsymbol{\Lambda}}_k$ in the neighborhood of \mathbf{u}_{∞} is

$$\delta \mathbf{\Lambda}_k = \mathsf{T}_\infty \delta \hat{\mathbf{\Lambda}}_k + \begin{pmatrix} \boldsymbol{\lambda}_\infty^{\mathbf{G}\top} \\ \mathbf{0} \end{pmatrix} \delta \mathbf{u}_k, \quad (4.49)$$

we have the first-order approximation

$$\delta \hat{\mathbf{\Lambda}}_k = -\mathsf{T}_\infty^{-1} \begin{pmatrix} \boldsymbol{\lambda}_\infty^{\mathbf{G}\top} \\ \mathbf{0} \end{pmatrix} \delta \mathbf{u}_k + \mathsf{T}_\infty^{-1} \boldsymbol{\Xi}_\infty \delta \mathbf{u}_k + o(\|\delta \mathbf{u}_k\|). \quad (4.50)$$

After replacement of (4.50) in (4.48) and rearrangement, we have

$$\delta \mathbf{u}_{k+1} = \boldsymbol{\Upsilon}_\infty^f \delta \mathbf{u}_k + o(\|\delta \mathbf{u}_k\|) \quad (4.51)$$

with

$$\boldsymbol{\Upsilon}_\infty^f := \mathbf{I} - \mathsf{K} \left(\mathbf{I} + \mathcal{P}_\mathbf{u} \mathsf{M}_\infty^{-1} \mathsf{N}_\infty \mathsf{T}_\infty^{-1} \left[\boldsymbol{\Xi}_\infty - \begin{pmatrix} \boldsymbol{\lambda}_\infty^{\mathbf{G}\top} \\ \mathbf{0} \end{pmatrix} \right] \right) \quad (4.52)$$

It follows that a necessary condition for the modifier-adaptation algorithm \mathcal{M}_f to converge to \mathbf{u}_∞ is that the gain matrix K be chosen such that the $(n_u \times n_u)$ matrix $\boldsymbol{\Upsilon}_\infty^f$ has spectral radius less than 1. Moreover, the relation (4.51) establishes the linear rate of convergence of the modifier-adaptation scheme. \square

Notice that the $(n_g + 3n_u) \times n_u$ matrix

$$\mathsf{N}_\infty^f := \mathsf{N}_\infty \mathsf{T}_\infty^{-1} \left[\begin{pmatrix} \boldsymbol{\lambda}_\infty^{\mathbf{G}} & \mathbf{0} \end{pmatrix}^\top - \boldsymbol{\Xi}_\infty \right]$$

is the Jacobian of (4.45) and (4.46) with respect to \mathbf{u} at \mathbf{u}_∞ . N_∞^f has the following form:

$$\mathsf{N}_\infty^f = \begin{pmatrix} \frac{\partial^2 \Phi_p}{\partial \mathbf{u}^2} - \frac{\partial^2 \Phi}{\partial \mathbf{u}^2} + \sum_{i=1}^{n_g} \mu_{i,\infty} \left(\frac{\partial^2 G_{p,i}}{\partial \mathbf{u}^2} - \frac{\partial^2 G_i}{\partial \mathbf{u}^2} \right) \\ -\mu_{1,\infty} \mathbf{u}_\infty^\top \left(\frac{\partial^2 G_{p,1}}{\partial \mathbf{u}^2} - \frac{\partial^2 G_1}{\partial \mathbf{u}^2} \right) \\ \vdots \\ -\mu_{n_g,\infty} \mathbf{u}_\infty^\top \left(\frac{\partial^2 G_{p,n_g}}{\partial \mathbf{u}^2} - \frac{\partial^2 G_{n_g}}{\partial \mathbf{u}^2} \right) \\ \mathbf{0}_{2n_u \times n_u} \end{pmatrix}_{(\mathbf{u}_\infty)}$$

Remark 4.2 In the special case where $\mathsf{K} = \mathbf{I}$, the matrix $\boldsymbol{\Upsilon}_\infty^f$ reduces to:

$$\mathbf{Y}_\infty^f = \mathcal{P}_u \mathbf{M}_\infty^{-1} \mathbf{N}_\infty \mathbf{T}_\infty^{-1} \left[\begin{pmatrix} \boldsymbol{\lambda}_\infty^G \\ \mathbf{0} \end{pmatrix} - \boldsymbol{\Xi}_\infty \right] \quad (4.53)$$

Notice that the $(n_K \times n_K)$ matrix

$$\mathbf{T}_\infty^{-1} \left[\begin{pmatrix} \boldsymbol{\lambda}_\infty^G \\ \mathbf{0} \end{pmatrix} - \boldsymbol{\Xi}_\infty \right] \mathcal{P}_u \mathbf{M}_\infty^{-1} \mathbf{N}_\infty$$

in Remark 4.1 has n_u eigenvalues equal to those of \mathbf{Y}_∞^f in (4.53), and $(n_K - n_u)$ eigenvalues equal to zero. This is consistent with the fact that, in the absence of filtering, the necessary conditions of Theorems 4.2 and 4.3 are equivalent.

Remark 4.3 (Unconstrained Problem) For an unconstrained optimization problem, the Jacobian matrices \mathbf{M}_∞ and \mathbf{N}_∞^f in Theorem 4.3 reduce to

$$\mathbf{M}_\infty = \frac{\partial^2 \Phi}{\partial \mathbf{u}^2}(\mathbf{u}_\infty, \boldsymbol{\theta}), \quad \mathbf{N}_\infty^f = \frac{\partial^2 \Phi_p}{\partial \mathbf{u}^2}(\mathbf{u}_\infty) - \frac{\partial^2 \Phi}{\partial \mathbf{u}^2}(\mathbf{u}_\infty, \boldsymbol{\theta}) \quad (4.54)$$

and condition (4.44) reduces to

$$\varrho \left\{ \mathbf{I} - \mathbf{K} \left(\frac{\partial^2 \Phi}{\partial \mathbf{u}^2} \right)_{(\mathbf{u}_\infty)}^{-1} \left(\frac{\partial^2 \Phi_p}{\partial \mathbf{u}^2} \right)_{(\mathbf{u}_\infty)} \right\} < 1, \quad (4.55)$$

Criterion 4.1 for model adequacy still holds for modifier-adaptation with filter on the inputs.

Alternative Modification

This second variant of the modifier-adaptation algorithm considers a linear modification of the process model rather than of the cost and constraint functions in the optimization problem. At a given operating point $\underline{\mathbf{u}}$, the process output is modified as:

$$\mathbf{y}_m(\mathbf{u}, \boldsymbol{\theta}) := \mathbf{y}(\mathbf{u}, \boldsymbol{\theta}) + \boldsymbol{\varepsilon}^y + \boldsymbol{\lambda}^y \mathbf{T}^T (\mathbf{u} - \underline{\mathbf{u}}), \quad (4.56)$$

where $\boldsymbol{\varepsilon}^y \in \mathbb{R}^{n_y}$ and $\boldsymbol{\lambda}^y \in \mathbb{R}^{n_u \times n_y}$ stand for the model modifiers.

In this modifier-adaptation variant, the operating point is updated from the repeated solution of the following optimization problem:

$$\begin{aligned}
\mathbf{u}_{k+1}^* &\in \arg \min_{\mathbf{u}} \quad \tilde{\Phi}_m(\mathbf{u}, \boldsymbol{\theta}) := \phi(\mathbf{u}, \mathbf{y}(\mathbf{u}, \boldsymbol{\theta})) + \boldsymbol{\varepsilon}_k^{\mathbf{y}} + \boldsymbol{\lambda}_k^{\mathbf{y}\top}(\mathbf{u} - \mathbf{u}_k) \\
\text{s.t.} \quad &\tilde{\mathbf{G}}_m(\mathbf{u}, \boldsymbol{\theta}) := \mathbf{g}(\mathbf{u}, \mathbf{y}(\mathbf{u}, \boldsymbol{\theta})) + \boldsymbol{\varepsilon}_k^{\mathbf{y}} + \boldsymbol{\lambda}_k^{\mathbf{y}\top}(\mathbf{u} - \mathbf{u}_k) \leq \mathbf{0} \\
&\mathbf{u}^{\text{L}} \leq \mathbf{u} \leq \mathbf{u}^{\text{U}}
\end{aligned}$$

where \mathbf{u}_k stands for the current operating point. Again, one can choose between two filtering strategies:

- 1) Filter the $n_y(n_u + 1)$ model modifiers $\boldsymbol{\varepsilon}^{\mathbf{y}}$ and $\boldsymbol{\lambda}^{\mathbf{y}}$, in which case the adaptation proceeds as:

$$\begin{aligned}
\varepsilon_{k+1}^{y_i} &:= (1 - b_i)\varepsilon_k^{y_i} + b_i [y_{p,i}(\mathbf{u}_{k+1}) - y_i(\mathbf{u}_{k+1}, \boldsymbol{\theta})], \quad i = 1, \dots, n_y \\
\boldsymbol{\lambda}_{k+1}^{y_i \top} &:= (1 - q_i)\boldsymbol{\lambda}_k^{y_i \top} + q_i \left[\frac{\partial y_{p,i}}{\partial \mathbf{u}}(\mathbf{u}_{k+1}) - \frac{\partial y_i}{\partial \mathbf{u}}(\mathbf{u}_{k+1}, \boldsymbol{\theta}) \right], \\
&i = 1, \dots, n_y,
\end{aligned}$$

with the gain coefficients $b_1, \dots, b_{n_y}, q_1, \dots, q_{n_y} \in (0, 1]$, while the new operating point is simply given by $\mathbf{u}_{k+1} := \mathbf{u}_{k+1}^*$.

- 2) Filter the n_u inputs \mathbf{u} , which results in the same adaptation law as (4.42) for the determination of the new operating point \mathbf{u}_{k+1} , while the model modifiers are calculated as:

$$\boldsymbol{\varepsilon}_k^{\mathbf{y}} := \mathbf{y}_p(\mathbf{u}_k) - \mathbf{y}(\mathbf{u}_k, \boldsymbol{\theta}) \quad (4.57)$$

$$\boldsymbol{\lambda}_k^{\mathbf{y}\top} := \frac{\partial \mathbf{y}_p}{\partial \mathbf{u}}(\mathbf{u}_k) - \frac{\partial \mathbf{y}}{\partial \mathbf{u}}(\mathbf{u}_k, \boldsymbol{\theta}). \quad (4.58)$$

Interestingly, the model modifiers (4.57) correspond to the model shift term \mathbf{a}_k introduced in Subsection 2.4.2.

If both $\phi(\mathbf{u}, \mathbf{y})$ and $\mathbf{g}(\mathbf{u}, \mathbf{y})$ are linear in \mathbf{u} and \mathbf{y} , the latter scheme is identical to the modifier-adaptation variant (4.43) with filter on the inputs. The difference between the two schemes is the definition of the modifiers and the way the approach is viewed, either as an approach where the modifiers are used to adapt the model that is subsequently used in the optimization step, or as an approach where the modifiers are used to adapt the constraints and the cost function in the optimization problem. It follows from these considerations that the local convergence condition for the modifier-adaptation variant with modifiers on the model outputs and filters on the inputs is identical to that given in Theorem 4.3 when the modifiers are on the cost and constraint functions.

4.1.7 Link to Previous Work

This subsection intends to highlight the distinction between the gradient modifiers described in this thesis, and the gradient modifiers proposed in the ISOPE literature.

Note that, if the gradient of the cost function is modified using the ISOPE modifier (2.34), condition (2.33) is required for the gradient of the modified cost function to match the plant gradient upon convergence. This is a bit awkward since, in principle, output value matching should not be a prerequisite for output gradient matching. This inconsistency can be removed by defining the gradient modifier λ^Φ as in Subsection 4.1.1,

$$\begin{aligned} \lambda_k^{\Phi\top} &:= \frac{\partial\Phi_p}{\partial\mathbf{u}}(\mathbf{u}_k) - \frac{\partial\Phi}{\partial\mathbf{u}}(\mathbf{u}_k, \boldsymbol{\theta}) \\ &= \frac{\partial\phi}{\partial\mathbf{u}}(\mathbf{u}_k, \mathbf{y}_p(\mathbf{u}_k)) + \frac{\partial\phi}{\partial\mathbf{y}}(\mathbf{u}_k, \mathbf{y}_p(\mathbf{u}_k)) \frac{\partial\mathbf{y}_p}{\partial\mathbf{u}}(\mathbf{u}_k) \\ &\quad - \frac{\partial\phi}{\partial\mathbf{u}}(\mathbf{u}_k, \mathbf{y}(\mathbf{u}_k, \boldsymbol{\theta})) - \frac{\partial\phi}{\partial\mathbf{y}}(\mathbf{u}_k, \mathbf{y}(\mathbf{u}_k, \boldsymbol{\theta})) \frac{\partial\mathbf{y}}{\partial\mathbf{u}}(\mathbf{u}_k, \boldsymbol{\theta}). \end{aligned} \quad (4.59)$$

Interestingly, if condition (2.33) is satisfied, the gradient modifier λ^Φ as defined in (4.59) reduces to the ISOPE modifier λ in (2.34).

For handling process-dependent constraints, the methodology presented in this thesis uses the approach of modifying the constraint functions introduced by Gao and Engell [37]. This approach differs significantly from that used in the ISOPE literature, which introduces a Lagrangian modifier in the cost function [10, 28, 94]. However, major differences also exist with the present work in that Gao and Engell [37] used some of the ISOPE features in their algorithm, such as the cost gradient modifier (including the model shift discussed in Subsection 2.4.2).

4.1.8 Case Study: Experimental Three-Tank System

The objective of this case study is to illustrate the applicability of the constraint adaptation and modifier adaptation approaches on a real experimental setup.

Experimental Setup

The three-tank system is depicted in Figure 4.4. It consists of the three plexiglass cylinders T1, T2 and T3, whose cross-section area and

height are $A = 154 \text{ cm}^2$ and $H = 60 \text{ cm}$, respectively. Each cylinder is equipped with a manual bottom valve V1, V2 and V3. Moreover, the cylinder are serially connected through the manual valves V12 and V32. Liquid (water) is fed to the cylinders T1 and T3 by the pumps P1 and P2, respectively, at a maximum flow rate of 6 L/min. Level measurements in the cylinders are carried out by piezo-resistive differential pressure sensors.

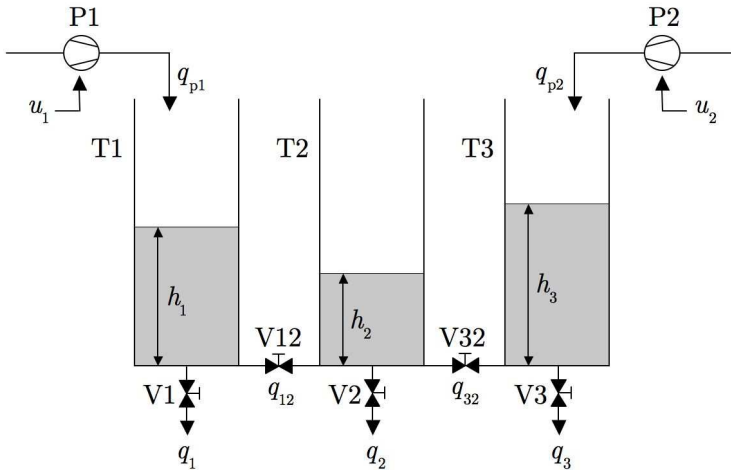


Fig. 4.4. Three-Tank System.

Subsequently, q_1 , q_2 and q_3 denote the outlet flow rates; q_{12} and q_{32} , the serial connection flow rates; q_{p1} and q_{p2} , the inlet flow rates delivered by the pumps; and h_1 , h_2 and h_3 , the levels in the cylinders.

The liquid level h_i in each tank is obtained as an affine function of the voltage signal y_i provided by the level sensor in that tank,

$$h_i = z_{i1} + z_{i2} y_i, \quad i = 1, \dots, 3.$$

The flow rate q_{pi} delivered by each pump is proportional to the voltage $u_i \in [0, 8]$ applied to that pump,

$$q_{pi} = w_i u_i, \quad i = 1, 2$$

The voltages u_1 and u_2 , rather than the flow rates q_{p1} and q_{p2} are considered as input variables. The numerical values for the coefficients z_{i1} , z_{i2} and w_i are reported in Table 4.2.

Table 4.2. Calibration coefficients for the levels in the tanks and the flow rates through the pumps.

	z_{i1}	z_{i2}		w_i
T1	37.69	-10.79	P1	13.22
T2	35.50	-7.224	P2	14.96
T3	40.94	-7.177		

Table 4.3. Calibration coefficients for the flow rates from the tanks.

	a_i		a_{j2}
T1	0.1203	T1-T2	0.0381
T2	0.0613	T2-T3	0.0285
T3	0.1141		

Optimization Problem Formulation

At steady-state, the mass conservation equations read:

$$A \frac{dh_1}{dt} = q_{p1} - q_1 - q_{12} = 0 \quad (4.60)$$

$$A \frac{dh_2}{dt} = q_{12} + q_{32} - q_2 = 0 \quad (4.61)$$

$$A \frac{dh_3}{dt} = q_{p2} - q_3 - q_{32} = 0. \quad (4.62)$$

For simplicity, the flow rates are modeled as functions of the tank levels using Torricelli's rule:

$$q_i = A a_i \sqrt{h_i}, \quad i = 1, \dots, 3 \quad (4.63)$$

$$q_{j2} = \text{sign}(d_{j2}) A a_{j2} \sqrt{|d_{j2}|}, \quad j = 1, 3, \quad (4.64)$$

with $d_{j2} := h_j - h_2$. The model parameters a_i , $i = 1, \dots, 3$, and a_{j2} , $j = 1, 3$, have been estimated from steady-state measurements using a least-square approach. The estimated values are reported in Table 4.3. This model is selected as the nominal model subsequently. The time constant of the system, with all the valves in fully open position, is about 3 minutes.

The optimization problem consists in minimizing the overall pumping effort, while maintaining the liquid level between given limits:

$$\begin{aligned}
& \min_{u_1, u_2} u_1^2 + u_2^2 & (4.65) \\
& \text{s.t.} \quad \text{model (4.60)-(4.64)} \\
& \quad h^L \leq h_1, h_2, h_3 \leq h^U, \\
& \quad u^L \leq u_1, u_2 \leq u^U.
\end{aligned}$$

Upper and lower bounds on the pump voltages and liquid levels are taken as $h^L = 5$ cm, $h^U = 30$ cm, $u^L = 0$ V and $u^U = 8$ V.

In the forthcoming implementations of modifier adaptation, the pump voltages do not become lower than 4 V, and the three-tank system operates always with $d_{12} > 0$ and $d_{32} > 0$. Hence, within the region of operation, the sign of d_{12} and d_{32} does not change, and (4.64) gives smooth equations. Therefore, the assumption made in Subsection 2.1.1 that the constraints \mathbf{G} are twice continuously differentiable with respect to \mathbf{u} holds.

Real-time Optimization using Modifier Adaptation

Modifier adaptation is implemented based on the update laws (3.11) and (4.12). Note that the cost gradient does not need adaptation since the cost function, which depends only on the input variables u_1 and u_2 , is perfectly known. A RTO period of 30 min is chosen, which leaves sufficient time for the system to reach steady-state after an input change. Measurements of the tank levels are taken every 1 sec, and the average value over the last 10 min of measurements is considered when adapting the modifiers.

Constraint Adaptation Alone.

The results shown in Figure 4.5 correspond to the modifier adaptation iterates with no correction of the constraint gradients. The algorithm starts from the conservative operating point $u_1 = 6.0$, $u_2 = 6.2$ and uses the constraint filter parameter $b = 0.5$.

In the first part of the experiment, all the manual valves are in fully open position. The iterates converge to an operating point where h_2 is at its lower bound of 5 cm. At time $t = 25.6$ h, a perturbation is introduced in the system by partially closing the valve V32 so as to reduce the flow rate q_{32} ; this time is indicated by a vertical dash-dotted line in Figure 4.5. It can be seen that it takes only a few iterations for constraint adaptation to reject this perturbation, and converge to a different operating point with again h_2 at its lower bound.

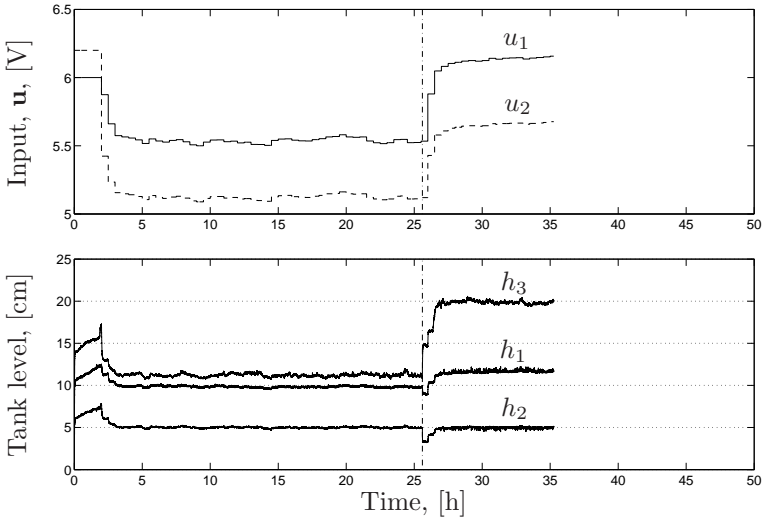


Fig. 4.5. Constraint adaptation alone applied to Problem (4.65).

Full Modifier Adaptation.

The results obtained by applying the modifier adaptation algorithm with correction of the constraint gradients are depicted in Figure 4.6. The iterates start from the same conservative operating point as before, and the update laws (3.11) and (4.12) are applied with $b = q = 0.5$. In order to estimate the experimental gradients, finite-difference perturbations are imposed on both inputs at each RTO iteration, with a perturbation amplitude of -0.5 V. These perturbations, which can be seen in the upper plot of Figure 4.6, call for an increase of the RTO period from 30 min to 90 min. At time $t = 26.24$ h, the same perturbation on the position of valve V32 is introduced.

In either part of the experiment, the iterates converge to an operating point for which the lower bound of h_2 is active. A slight violation of the constraint on h_2 is observed at each iteration due to the finite-difference scheme for gradient estimation. Note also that, similar to the case without gradient correction, the constraint on h_2 is violated during the transient after the valve perturbation.

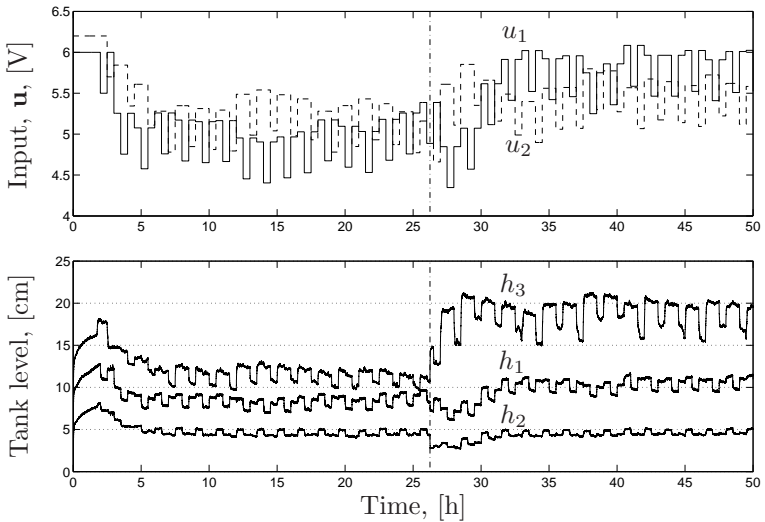


Fig. 4.6. Full modifier adaptation applied to Problem (4.65).

Comparison.

A comparison in terms of the cost value at the RTO iterations without and with constraint gradient correction is given in Figure 4.7. For easier comparison, the time coordinates are shifted so that, in both experiments, the origin corresponds to the times at which the perturbation in valve V32 is applied. As expected, modifier adaptation with gradient correction yields a lower cost value upon convergence. In Figure 4.7 only the cost obtained at the RTO iterations is represented in the case of full modifier adaptation since, at the finite-difference perturbations, the constraint on h_2 is violated. This violation could be avoided if the perturbation amplitude was selected as 0.5 V, instead of -0.5 V. However, in this case the cost corresponding to the finite difference perturbations would be higher than that obtained with constraint adaptation alone, making the overall cost for the full modifier adaptation worse than that for constraint adaptation alone. This is a typical situation where there is little to gain by correcting the gradients. In such a situation, it is not recommended to update the gradient modifiers at each RTO iteration, but at a slower rate.

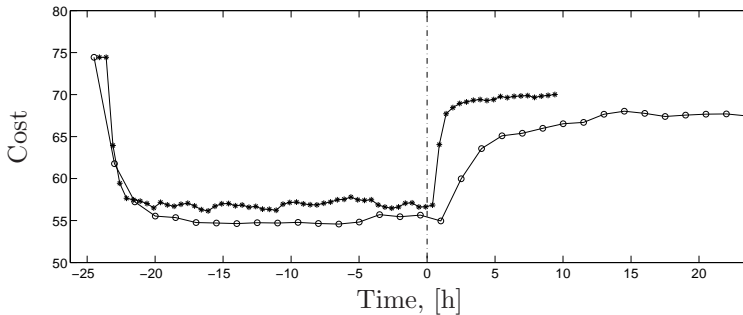


Fig. 4.7. Cost comparison. $-*-\cdot$: constraint adaptation; $-o-$: full modifier adaptation.

4.2 Dual Modifier Adaptation

The FFD approach for estimating experimental gradients was presented in Subsection 2.4.3. It consists in perturbing each input individually around the current operating point to get an estimate of the corresponding gradient element. This is done at each RTO iteration. An alternative approach proposed in [12, 13] in the context of the dual ISOPE algorithm is to estimate the gradients based on the current and past operating points rather than perturbing each input individually around the current operating point (see Subsection 2.4.4). In dual ISOPE, a constraint that is representative of the ill-conditioning that might occur due to the relative location of the successive operating points generated by RTO is added to the RTO optimization problem. However, this constraint is not directly related to the errors in the estimated gradients. Subsection 4.2.1 presents an analysis of the error in the estimated gradient of a general noisy function, and a new constraint is defined that enforces an upper bound on the gradient error. In the case of unconstrained optimization problems, this constraint is applied to the cost function and is introduced into a dual modifier-adaptation algorithm in Subsection 4.2.2. Strategies on how to apply this constraint in the case of constrained optimization problems are discussed in Subsection 4.2.3.

4.2.1 Estimation of Experimental Gradient from Past Operating Points

Gradient values depend on the order of magnitude of the decision variables \mathbf{u} . It is assumed throughout this section that all the decision variables \mathbf{u} are of the same order of magnitude, which can be achieved via scaling. For example, if the decision variable u_i remains within the interval $[u_{i,a}, u_{i,b}]$, it can be scaled as $u_i^{\text{scaled}} = (u_i - u_{i,a}) / (u_{i,b} - u_{i,a})$. For notational simplicity, the superscript indicating a scaled variable will be omitted in the sequel.

The analysis in this section is carried out for a general noisy function of the form

$$\psi(\mathbf{u}) = \Psi(\mathbf{u}) + v, \quad (4.66)$$

where v represents the measurement noise. The forthcoming analysis is conducted assuming that the noisy function $\psi(\mathbf{u})$ remains within an interval δ at steady-state operation, as illustrated in Figure 4.8. Based on a statistical description of v , δ could be selected by considering a desired confidence interval. Values that fall outside the selected confidence interval could simply be discarded.

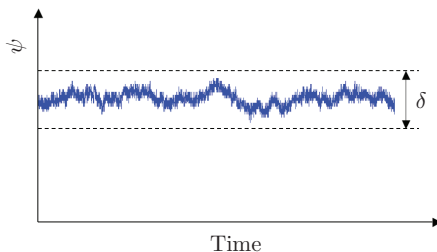


Fig. 4.8. Noise at steady state operation.

Using a first-order approximation of Ψ in the neighborhood of \mathbf{u} , the value of ψ at the current and past operating points \mathbf{u}_{k-j} , $j = 0, \dots, n_u - 1$, is given by

$$\begin{aligned} \psi(\mathbf{u}_{k-j}) &= \psi(\mathbf{u}) + \frac{\partial \Psi}{\partial \mathbf{u}}(\mathbf{u})[\mathbf{u}_{k-j} - \mathbf{u}] + O(\|\mathbf{u}_{k-j} - \mathbf{u}\|^2) \quad (4.67) \\ &+ v_{k-j} - v, \quad j = 0, \dots, n_u - 1. \end{aligned}$$

An estimate $\hat{\beta}$ of the experimental gradient $\frac{\partial \Psi}{\partial \mathbf{u}}(\mathbf{u})$ at the operating point \mathbf{u} satisfies

$$\psi(\mathbf{u}_{k-j}) = \psi(\mathbf{u}) + \hat{\beta}[\mathbf{u}_{k-j} - \mathbf{u}], \quad j = 0, \dots, n_u - 1 \quad (4.68)$$

That is, given the n_u operating points, $\mathbf{u}_k, \mathbf{u}_{k-1}, \dots, \mathbf{u}_{k-n_u+1}$, and the corresponding measured function values $\psi(\mathbf{u}_k), \psi(\mathbf{u}_{k-1}), \dots, \psi(\mathbf{u}_{k-n_u+1})$, $\hat{\beta}$ can be obtained by writing (4.68) in the following matrix form [13]:

$$\hat{\beta}(\mathbf{u}) = \mathcal{Y}(\mathbf{u}) \mathcal{U}^{-1}(\mathbf{u}) \quad (4.69)$$

with

$$\mathcal{U}(\mathbf{u}) := [\mathbf{u} - \mathbf{u}_k \ \dots \ \mathbf{u} - \mathbf{u}_{k-n_u+1}] \in \mathbb{R}^{n_u \times n_u} \quad (4.70)$$

$$\mathcal{Y}(\mathbf{u}) := [\psi(\mathbf{u}) - \psi(\mathbf{u}_k) \ \dots \ \psi(\mathbf{u}) - \psi(\mathbf{u}_{k-n_u+1})] \in \mathbb{R}^{1 \times n_u}. \quad (4.71)$$

In turn, the gradient error, $\epsilon(\mathbf{u}) := \hat{\beta}(\mathbf{u}) - \frac{\partial \Psi}{\partial \mathbf{u}}(\mathbf{u})$, can be written as $\epsilon(\mathbf{u}) = \epsilon^t(\mathbf{u}) + \epsilon^n(\mathbf{u})$, with:

$$\begin{aligned} \epsilon^t(\mathbf{u}) &= [\Psi(\mathbf{u}) - \Psi(\mathbf{u}_k) \ \dots \ \Psi(\mathbf{u}) - \Psi(\mathbf{u}_{k-n_u+1})] \mathcal{U}^{-1}(\mathbf{u}) \\ &\quad - \frac{\partial \Psi}{\partial \mathbf{u}}(\mathbf{u}) \end{aligned} \quad (4.72)$$

$$\epsilon^n(\mathbf{u}) = [v - v_k \ \dots \ v - v_{k-n_u+1}] \mathcal{U}^{-1}(\mathbf{u}), \quad (4.73)$$

where ϵ^t and ϵ^n represent errors due to truncation and measurement noise, respectively. The truncation error is due to the approximation incurred by using finite differences and is related to the curvature of $\Psi(\mathbf{u})$. The measurement noise error is due to the presence of measurement noise in $\psi(\mathbf{u})$.

It is assumed in this analysis that the $n_u + 1$ operating points in $\mathcal{U}(\mathbf{u})$ are such that $\mathcal{U}(\mathbf{u})$ is invertible.

Gradient Error due to Truncation

An upper bound on the norm of the gradient error due to truncation is given in the next proposition.

Proposition 4.1 (Upper Bound on Gradient Error due to Truncation) *Assume that $\Psi(\mathbf{u})$ is twice continuously differentiable with respect to \mathbf{u} . Given the current and past operating points, $\mathbf{u}_k, \mathbf{u}_{k-1}, \dots, \mathbf{u}_{k-n_u+1}$, then, at the new operating point \mathbf{u} , an upper bound on $\|\epsilon^t(\mathbf{u})\|$ is given by*

$$\|\boldsymbol{\epsilon}^t(\mathbf{u})\| \leq \mathcal{E}^t(\mathbf{u}), \quad (4.74)$$

where

$$\begin{aligned} \mathcal{E}^t(\mathbf{u}) := \frac{d_2}{2} \left\| \left[(\mathbf{u} - \mathbf{u}_k)^\top (\mathbf{u} - \mathbf{u}_k) \dots \right. \right. \\ \left. \left. \dots (\mathbf{u} - \mathbf{u}_{k-n_u+1})^\top (\mathbf{u} - \mathbf{u}_{k-n_u+1}) \right] \mathcal{U}^{-1}(\mathbf{u}) \right\| \end{aligned} \quad (4.75)$$

with d_2 the largest absolute eigenvalue of the Hessian of $\Psi(\cdot)$.

Proof. Applying Taylor's theorem in the neighborhood of \mathbf{u} gives:

$$\begin{aligned} \Psi(\mathbf{u}) - \bar{\Psi}(\mathbf{u}_{k-j}) &= \frac{\partial \Psi}{\partial \mathbf{u}}(\mathbf{u})(\mathbf{u} - \mathbf{u}_{k-j}) \\ &\quad - \frac{1}{2}(\mathbf{u} - \mathbf{u}_{k-j})^\top \frac{\partial^2 \Psi}{\partial \mathbf{u}^2}(\bar{\mathbf{u}}_j)(\mathbf{u} - \mathbf{u}_{k-j}), \end{aligned} \quad (4.76)$$

with $\bar{\mathbf{u}}_j = \mathbf{u} + \vartheta_j(\mathbf{u}_{k-j} - \mathbf{u})$, $j = 0, \dots, n_u - 1$, for some $\vartheta_j \in [0, 1]$. Substituting (4.76) into (4.72) then gives:

$$\begin{aligned} \boldsymbol{\epsilon}^t(\mathbf{u}) &= \left(\frac{\partial \Psi}{\partial \mathbf{u}}(\mathbf{u}) \mathcal{U}(\mathbf{u}) - \frac{1}{2} \left[(\mathbf{u} - \mathbf{u}_k)^\top \frac{\partial^2 \Psi}{\partial \mathbf{u}^2}(\bar{\mathbf{u}}_0)(\mathbf{u} - \mathbf{u}_k) \dots \right. \right. \\ &\quad \left. \left. \dots (\mathbf{u} - \mathbf{u}_{k-n_u+1})^\top \frac{\partial^2 \Psi}{\partial \mathbf{u}^2}(\bar{\mathbf{u}}_{n_u-1})(\mathbf{u} - \mathbf{u}_{k-n_u+1}) \right] \right) \mathcal{U}^{-1}(\mathbf{u}) - \frac{\partial \Psi}{\partial \mathbf{u}}(\mathbf{u}) \\ &= -\frac{1}{2} \left[(\mathbf{u} - \mathbf{u}_k)^\top \frac{\partial^2 \Psi}{\partial \mathbf{u}^2}(\bar{\mathbf{u}}_0)(\mathbf{u} - \mathbf{u}_k) \dots \right. \\ &\quad \left. \dots (\mathbf{u} - \mathbf{u}_{k-n_u+1})^\top \frac{\partial^2 \Psi}{\partial \mathbf{u}^2}(\bar{\mathbf{u}}_{n_u-1})(\mathbf{u} - \mathbf{u}_{k-n_u+1}) \right] \mathcal{U}^{-1}(\mathbf{u}). \end{aligned} \quad (4.77)$$

The Hessian matrix $\mathbf{H}_j := \frac{\partial^2 \Psi}{\partial \mathbf{u}^2}(\bar{\mathbf{u}}_j)$ being symmetric, there exists an orthogonal matrix \mathbf{P}_j such that $\mathbf{P}_j^\top \mathbf{H}_j \mathbf{P}_j = \mathbf{D}_j = \text{diag}(\lambda_{1,j}, \dots, \lambda_{n_u,j})$, where $\lambda_{i,j} \in \sigma(\mathbf{H}_j)$, and $\sigma(\cdot)$ stands for the spectrum of a matrix. Letting $\mathbf{w}_j = \mathbf{P}_j^\top (\mathbf{u} - \mathbf{u}_{k-j})$ gives:

$$\begin{aligned} |(\mathbf{u} - \mathbf{u}_{k-j})^\top \mathbf{H}_j (\mathbf{u} - \mathbf{u}_{k-j})| &= |\mathbf{w}_j^\top \mathbf{P}_j^\top \mathbf{H}_j \mathbf{P}_j \mathbf{w}_j| \\ &= |\mathbf{w}_j^\top \mathbf{D}_j \mathbf{w}_j| = \left| \sum_{i=1}^{n_u} \lambda_{i,j} w_{i,j}^2 \right| \leq d_2 \sum_{i=1}^{n_u} w_{i,j}^2 = d_2 \mathbf{w}_j^\top \mathbf{w}_j \\ &= d_2 (\mathbf{u} - \mathbf{u}_{k-j})^\top (\mathbf{u} - \mathbf{u}_{k-j}), \quad j = 0, \dots, n_u - 1. \end{aligned} \quad (4.78)$$

Let us introduce the positive-definite matrix $\mathbf{R} := \mathcal{U}^{-1}(\mathbf{u}) [\mathcal{U}^{-1}(\mathbf{u})]^\top$ and the vectors

$$\begin{aligned} \mathbf{p}_1^\top &:= [\mathbf{w}_0^\top D_0 \mathbf{w}_0 \ \dots \ \mathbf{w}_{n_u-1}^\top D_{n_u-1} \mathbf{w}_{n_u-1}], \\ \mathbf{p}_2^\top &:= [\mathbf{d}_2 \mathbf{w}_0^\top \mathbf{w}_0 \ \dots \ \mathbf{d}_2 \mathbf{w}_{n_u-1}^\top \mathbf{w}_{n_u-1}]. \end{aligned}$$

It follows from (4.78) that the absolute value of each element of \mathbf{p}_1 is less than or equal to the corresponding element of \mathbf{p}_2 , and thus $\mathbf{p}_1^\top \mathbf{R} \mathbf{p}_1 \leq \mathbf{p}_2^\top \mathbf{R} \mathbf{p}_2$, from where (4.74) follows. \square

Note that \mathbf{d}_2 represents an upper bound on the curvature of $\Psi(\cdot)$.

Claim 4.1 *If in Proposition 4.1 the points are taken as $\mathbf{u}_{k-j} = \mathbf{u} + h\mathbf{e}_{j+1}$, $j = 0, \dots, n_u - 1$, there results the FFD arrangement with $h > 0$ being the step size and \mathbf{e}_{j+1} the $(j + 1)$ st unit vector. In this particular case, it can be shown that (4.74) reduces to*

$$\|\boldsymbol{\epsilon}^t\| \leq \frac{\mathbf{d}_2}{2} \sqrt{n_u} h, \quad (4.79)$$

which is no longer a function of \mathbf{u} .

Proof. It follows from $\mathbf{u}_{k-j} = \mathbf{u} + h\mathbf{e}_{j+1}$ that $(\mathbf{u} - \mathbf{u}_{k-j})^\top (\mathbf{u} - \mathbf{u}_{k-j}) = h^2$, $j = 0, \dots, n_u - 1$, and $\mathcal{U}(\mathbf{u}) = \text{diag}(-h, -h, \dots, -h)$. Therefore

$$\mathcal{E}^t(\mathbf{u}) = \frac{\mathbf{d}_2}{2} \left\| [h^2 \ h^2 \ \dots \ h^2] \text{diag}\left(-\frac{1}{h}, -\frac{1}{h}, \dots, -\frac{1}{h}\right) \right\| = \frac{\mathbf{d}_2}{2} \sqrt{n_u} h.$$

and (4.79) follows from Proposition 4.1. \square

Notice that (4.79) is the same expression as that reported in [14] for the FFD approach.

Gradient Error due to Measurement Noise

A geometrical characterization of the vector $\boldsymbol{\epsilon}^n(\mathbf{u})$ is given first. Let the vector $\mathbf{n}_x \in \mathbf{R}^{n_u+1}$ be normal to the n_u -dimensional surface generated by the points $[\mathbf{u}^\top \ v]^\top$, $[\mathbf{u}_k^\top \ v_k]^\top$, \dots , $[\mathbf{u}_{k-n_u+1}^\top \ v_{k-n_u+1}]^\top$:

$$\begin{bmatrix} u_1 - u_{1,k} & \dots & u_{n_u} - u_{n_u,k} & v - v_k \\ u_1 - u_{1,k-1} & \dots & u_{n_u} - u_{n_u,k-1} & v - v_{k-1} \\ \vdots & & \vdots & \vdots \\ u_1 - u_{1,k-n_u+1} & \dots & u_{n_u} - u_{n_u,k-n_u+1} & v - v_{k-n_u+1} \end{bmatrix} \mathbf{n}_x = \mathbf{0} \quad (4.80)$$

The normal vector $\mathbf{n}_x \in \mathbb{R}^{n_u+1}$ can be obtained, e.g., by singular value decomposition of the $n_u \times (n_u + 1)$ matrix on the left side of (4.80), which is assumed to be of rank n_u . Next, we define \mathbf{n}_v as the unit vector that is normal to the points $[\mathbf{u}^\top \ 0]^\top, \dots, [\mathbf{u}_{k-n_u+1}^\top \ 0]^\top$, i.e., $\mathbf{n}_v = [0 \ \dots \ 0 \ 1]^\top$. The angle α between \mathbf{n}_x and \mathbf{n}_v is given by

$$\alpha = \arccos\left(\frac{\mathbf{n}_v^\top \mathbf{n}_x}{\|\mathbf{n}_v\| \|\mathbf{n}_x\|}\right) \quad (4.81)$$

In particular, by dividing each element of the vector \mathbf{n}_x by the last one, \mathbf{n}_x can be chosen as $\mathbf{n}_x = [\bar{\mathbf{n}}^\top \ 1]^\top$, $\bar{\mathbf{n}} \in \mathbb{R}^{n_u}$. With this

$$\alpha = \arccos\left(\frac{1}{\sqrt{\bar{\mathbf{n}}^\top \bar{\mathbf{n}} + 1}}\right) \quad (4.82)$$

and from (4.80)

$$\mathbf{U}(\mathbf{u})^\top \bar{\mathbf{n}} = -[v - v_k \quad v - v_{k-1} \quad \dots \quad v - v_{k-n_u+1}]^\top.$$

From (4.73), $\bar{\mathbf{n}}^\top = -\epsilon^n$, and (4.82) gives

$$\alpha = \arccos\left(\frac{1}{\sqrt{\|\epsilon^n\|^2 + 1}}\right) \quad (4.83)$$

The vectors \mathbf{n}_v , \mathbf{n}_x and ϵ^n are represented in Figure 4.9 for the two-dimensional case ($n_u = 2$). Independently of the number of inputs, a plane \mathcal{P} can be defined, which contains the three vectors \mathbf{n}_v , \mathbf{n}_x and ϵ^n . Notice that ϵ^n belongs to the input space, so its component in the direction of v is always zero. The relation (4.83), which is represented in Figure 4.10, takes place on the plane \mathcal{P} .

For relating the error norm $\|\epsilon^n(\mathbf{u})\|$ to the location of the new operating point as expressed in the matrix $\mathbf{U}(\mathbf{u})$, two definitions are required.

Definition 4.1 (Distance Between Complement Affine Subspaces) *Given a set of $(n_u + 1)$ points in a n_u -dimensional space, $\mathcal{S} := \{\mathbf{u}, \mathbf{u}_k, \dots, \mathbf{u}_{k-n_u+1}\}$, a proper subset of \mathcal{S} , $\mathcal{S}^A \subsetneq \mathcal{S}$ of $n_u^A \in \{1, \dots, n_u\}$ points, and its complement $\mathcal{S}^C := \mathcal{S} \setminus \mathcal{S}^A$ of $(n_u + 1 - n_u^A)$ points, the distance between complement affine subspaces is defined as the (orthogonal) distance between the affine subspace of dimension $(n_u^A - 1)$ generated by all the points in \mathcal{S}^A , and the affine subspace of dimension $(n_u - n_u^A)$ generated by all the points in \mathcal{S}^C .*

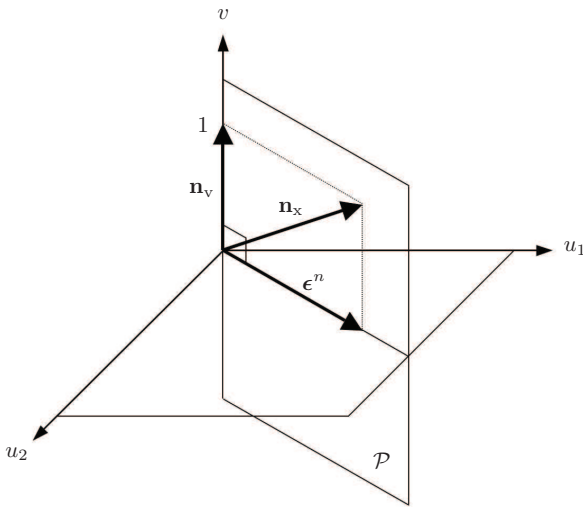


Fig. 4.9. Geometrical representation of ϵ^n in the two-dimensional case.

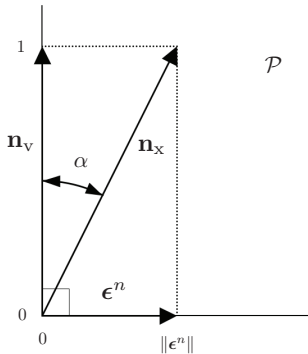


Fig. 4.10. Geometrical representation of ϵ^n on the plane \mathcal{P} .

The total number of possible pairs of complement affine subspaces that can be generated from \mathcal{S} is $n_b = 1 + \sum_{s=1}^{n_u-1} 2^s$. The way these distances can be calculated is described in Appendix B, where a definition of affine subspaces is also provided.

Definition 4.2 (Nearest Complement Affine Subspaces) *The shortest distance between complement affine subspaces is given by*

$l_{\min} := \min\{l_1, l_2, \dots, l_{n_b}\}$, where l_1, l_2, \dots, l_{n_b} are the distances between all possible pairs of complement affine subspaces that can be generated from \mathcal{S} .

Notice that, if any pair of complement affine subspaces intersect, then all pairs of subspaces intersect and $l_1 = l_2 = \dots = l_{n_b} = 0$.

In the 2-dimensional case ($n_u = 2$), the number of distances to evaluate is $n_b = 3$, which corresponds to the 3 point-to-line distances. In the 3-dimensional case, there are $n_b = 7$ distances to evaluate, which correspond to 4 point-to-plane distances, and 3 line-to-line distances.

The largest possible value of $\|\epsilon^n(\mathbf{u})\|$, noted $\|\epsilon^n(\mathbf{u})\|_{\max}$, is computed in the next proposition.

Proposition 4.2 (Upper Bound on Gradient Error due to Measurement Noise) *Given a set \mathcal{S} of $(n_u + 1)$ points in a n_u -dimensional space, $\mathcal{S} := \{\mathbf{u}, \mathbf{u}_k, \dots, \mathbf{u}_{k-n_u+1}\}$, and the interval δ for the noisy function ψ , the maximal value of $\|\epsilon^n(\mathbf{u})\|$ occurs when, for the subsets \mathcal{S}^A and \mathcal{S}^C corresponding to the nearest complement affine subspaces, the measurement error is $\delta/2$ for all the points in \mathcal{S}^A and $-\delta/2$ for all the points in \mathcal{S}^C (or $-\delta/2$ and $\delta/2$, respectively). Furthermore, $\|\epsilon^n(\mathbf{u})\|_{\max} = \delta/l_{\min}(\mathbf{u})$.*

Proof. The proof includes two steps.

1. For noise in the range $[-\delta/2, \delta/2]$, the worst case corresponds to a global optimum of the following optimization problem:

$$\begin{aligned} \max_{\mathbf{v}} \quad & \|\epsilon^n(\mathbf{u})\|^2 & (4.84) \\ \text{s.t.} \quad & -\delta/2 \leq v \leq \delta/2, \quad -\delta/2 \leq v_{k-j} \leq \delta/2, \quad j = 0, \dots, n_u - 1 \end{aligned}$$

with the decision variables $\mathbf{v} = [v \ v_k \ \dots \ v_{k-n_u+1}]^T$.

Using the matrix $\mathbf{R} = \mathcal{U}^{-1}(\mathbf{u}) [\mathcal{U}^{-1}(\mathbf{u})]^T$, the squared error norm can be expressed as

$$\begin{aligned} \|\epsilon^n(\mathbf{u})\|^2 = & [v \ \dots \ v] \mathbf{R} [v \ \dots \ v]^T & (4.85) \\ & - 2 [v \ \dots \ v] \mathbf{R} [v_k \ \dots \ v_{k-n_u+1}]^T \\ & + [v_k \ \dots \ v_{k-n_u+1}] \mathbf{R} [v_k \ \dots \ v_{k-n_u+1}]^T, \end{aligned}$$

or, equivalently,

$$\begin{aligned} \|\epsilon^n(\mathbf{u})\|^2 = & \eta_s v^2 - 2 v [\eta_1 \ \dots \ \eta_{n_u}] [v_k \ \dots \ v_{k-n_u+1}]^T & (4.86) \\ & + [v_k \ \dots \ v_{k-n_u+1}] \mathbf{R} [v_k \ \dots \ v_{k-n_u+1}]^T, \end{aligned}$$

where

$$\eta_s = \sum_{i=1}^{n_u} \sum_{j=1}^{n_u} r_{ij}, \quad \text{and} \quad \eta_j = \sum_{i=1}^{n_u} r_{ij}, \quad j = 1, \dots, n_u$$

Equation (4.86) can be written as:

$$\begin{aligned} \|\epsilon^n(\mathbf{u})\|^2 &= \mathbf{v}^\top \begin{bmatrix} \eta_s & \mathbf{0}_{1 \times n_u} \\ \mathbf{0}_{n_u \times 1} & \mathbf{R} \end{bmatrix} \mathbf{v} - 2 \mathbf{v}^\top \begin{bmatrix} 0 & \eta_1 & \dots & \eta_{n_u} \\ \mathbf{0}_{n_u \times 1} & \mathbf{0}_{n_u \times 1} & \dots & \mathbf{0}_{n_u \times 1} \end{bmatrix} \mathbf{v} \\ &= \mathbf{v}^\top \mathbf{A} \mathbf{v}, \quad \text{with} \quad \mathbf{A} := \begin{bmatrix} \eta_s & -2 \eta_1 & \dots & -2 \eta_{n_u} \\ \mathbf{0}_{n_u \times 1} & & & \mathbf{R} \end{bmatrix} \end{aligned}$$

The non-symmetric matrix \mathbf{A} is the sum of the symmetric matrix $\mathbf{M} = \frac{1}{2}(\mathbf{A} + \mathbf{A}^\top)$ and the skew symmetric matrix $\mathbf{C} = \frac{1}{2}(\mathbf{A} - \mathbf{A}^\top)$. For a skew symmetric matrix, $\mathbf{v}^\top \mathbf{C} \mathbf{v} = 0$, and therefore $\mathbf{v}^\top \mathbf{A} \mathbf{v} = \mathbf{v}^\top \mathbf{M} \mathbf{v}$. Hence, the optimization problem (4.84) can be reformulated as:

$$\begin{aligned} \max_{\mathbf{v}} \quad & \mathbf{v}^\top \mathbf{M} \mathbf{v} & (4.87) \\ \text{s.t.} \quad & -\delta/2 \leq v \leq \delta/2, \quad -\delta/2 \leq v_{k-j} \leq \delta/2, \quad j = 0, \dots, n_u - 1 \end{aligned}$$

with \mathbf{M} a symmetric matrix. It is well known that a global optimum of problem (4.87) must be attained at the intersection of the input bounds (see e.g. Theorem 32.1 in [75]). That is, all components in \mathbf{v} must be at their upper or lower bounds.

2. Next, we show that, if for any pair of complement affine subspaces, all the points in \mathcal{S}^A have error $v^A = \delta/2$ and all the points in \mathcal{S}^C have error $v^C = -\delta/2$ (or $-\delta/2$ and $\delta/2$ respectively), then the error vector $\epsilon^n(\mathbf{u})$ is normal to both affine subspaces. Considering the sets $\mathcal{S}^A = \{\mathbf{u}^1, \dots, \mathbf{u}^{n_u^A}\}$ and $\mathcal{S}^C = \{\mathbf{u}^{n_u^A+1}, \dots, \mathbf{u}^{n_u+1}\}$ ($\mathcal{S}^A \subsetneq \mathcal{S}$ and $\mathcal{S}^C := \mathcal{S} \setminus \mathcal{S}^A$), $\epsilon^n(\mathbf{u})$ in (4.73) can also be written as:

$$\epsilon^n(\mathbf{u}) = [0 \ \dots \ 0 \ v^A - v^C] \mathcal{U}_{AC}^{-1}(\mathbf{u}) \quad (4.88)$$

with

$$\begin{aligned} \mathcal{U}_{AC}(\mathbf{u}) := & [\mathbf{u}^1 - \mathbf{u}^2 \ \dots \ \mathbf{u}^1 - \mathbf{u}^{n_u^A} \ \mathbf{u}^{n_u^A+1} - \mathbf{u}^{n_u^A+2} \ \dots \\ & \mathbf{u}^{n_u^A+1} - \mathbf{u}^{n_u+1} \ \mathbf{u}^{n_u^A} - \mathbf{u}^{n_u^A+1}] \in \mathbb{R}^{n_u \times n_u} \end{aligned}$$

Using the matrix \mathbf{U} defined in (B.4) in $\mathcal{U}_{AC}(\mathbf{u})$, (4.88) becomes:

$$\epsilon^n(\mathbf{u}) \begin{bmatrix} \mathbf{U} \\ (\mathbf{u}^{n_u^A})^\top - (\mathbf{u}^{n_u^A+1})^\top \end{bmatrix}^\top = \begin{bmatrix} \mathbf{0}_{n_u-1} \\ v^A - v^C \end{bmatrix}^\top \quad (4.89)$$

from where it is clear that $\boldsymbol{\epsilon}^n(\mathbf{u})$ is normal to both complement affine subspaces.

From (4.89) and (B.5), one has

$$|\boldsymbol{\epsilon}^n(\mathbf{u})(\mathbf{u}^{n_u^A} - \mathbf{u}^{n_u^A+1})| = l_{AC} \|\boldsymbol{\epsilon}^n(\mathbf{u})\| = |v^A - v^C| = \delta, \quad (4.90)$$

where l_{AC} is the distance between the two complement affine subspaces. From all possible complement subsets \mathcal{S}^A and \mathcal{S}^C , $\|\boldsymbol{\epsilon}^n(\mathbf{u})\|_{\max}$ occurs for the complement subsets that correspond to the nearest complement affine subspaces, i.e. for $l_{AC} = l_{\min}(\mathbf{u})$. It follows that $\|\boldsymbol{\epsilon}^n(\mathbf{u})\|_{\max} = \frac{\delta}{l_{\min}(\mathbf{u})}$. \square

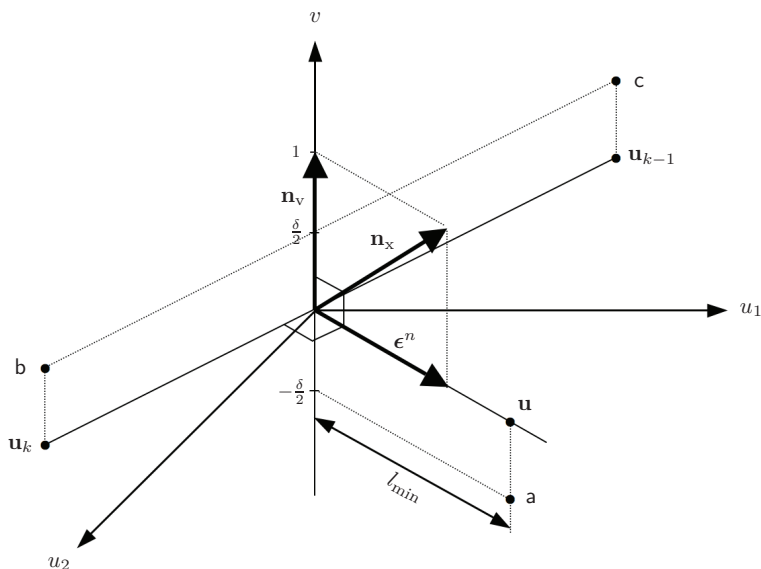


Fig. 4.11. Geometrical representation of worst-case scenario for measurement noise in the two-dimensional case.

The limiting situation given by Proposition 4.2 is represented in Figure 4.11 for the two-dimensional case ($n_u = 2$). There are three distances between complement affine subspaces that can be evaluated; say l_1 : the distance between \mathbf{u}_k and the line generated by \mathbf{u} and \mathbf{u}_{k-1} ; l_2 : the distance between \mathbf{u}_{k-1} and the line generated by \mathbf{u}_k and \mathbf{u} ;

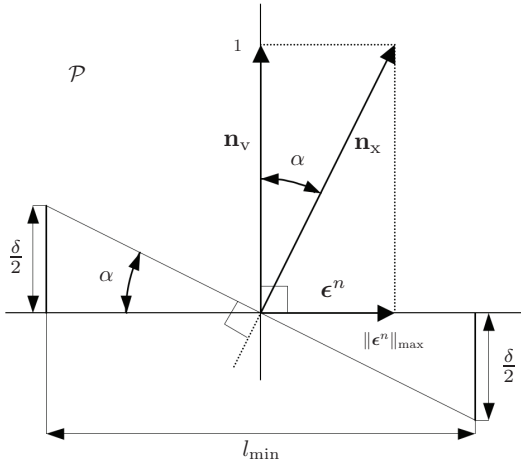


Fig. 4.12. Geometrical representation of worst-case scenario for measurement noise on the plane \mathcal{P} .

and l_3 : the distance between \mathbf{u} and the line generated by \mathbf{u}_k and \mathbf{u}_{k-1} . Consider the case where the current and past operating points \mathbf{u}_k and \mathbf{u}_{k-1} belong to the subset \mathcal{S}^A , while the new operating point \mathbf{u} belongs to \mathcal{S}^C . The operating points in these subsets generate the nearest complement affine subspaces, thus $l_{\min} = l_3 = \min\{l_1, l_2, l_3\}$.

When the measurement error is $\delta/2$ for all the points in \mathcal{S}^A and $-\delta/2$ for all the points in \mathcal{S}^C , ϵ^n is normal to both complement affine subspaces. According to Proposition 4.2, this situation leads to the norm of ϵ^n taking its largest possible value $\|\epsilon^n\|_{\max}$. Recall that the norm of \mathbf{n}_v is 1 by definition and vector \mathbf{n}_x is normal to the plane generated by the points \mathbf{a} , \mathbf{b} and \mathbf{c} (see (4.80)).

Independently of the number of inputs, a plane \mathcal{P} can be defined, which contains the three vectors \mathbf{n}_v , \mathbf{n}_x and ϵ^n . The representation of the worst-case scenario of Proposition 4.2 on the plane \mathcal{P} is represented in Figure 4.12. Notice that $\delta = l_{\min} \tan(\alpha)$, and $\|\epsilon^n\|_{\max} = \tan(\alpha)$. Hence, it is verified that $\|\epsilon^n\|_{\max} = \frac{\delta}{l_{\min}}$.

Remark 4.4 Recall that in order to evaluate $l_{\min}(\mathbf{u})$ it is necessary to evaluate all the distances between complement affine subspaces. Let us consider the distance between the new operating point \mathbf{u} and the hyperplane generated by the current and past operating points. In this

case, we have $\mathcal{S}^A := \{\mathbf{u}_k, \mathbf{u}_{k-1}, \dots, \mathbf{u}_{k-n_u+1}\}$ and $\mathcal{S}^C := \{\mathbf{u}\}$. The matrix $\mathbf{U} \in \mathbb{R}^{(n_u-1) \times n_u}$ defined in (B.4) is given by:

$$\mathbf{U}_k := [\mathbf{u}_k - \mathbf{u}_{k-1} \quad \mathbf{u}_k - \mathbf{u}_{k-2} \quad \dots \quad \mathbf{u}_k - \mathbf{u}_{k-n_u+1}]^\top$$

Denoting by \mathbf{n}_k the vector normal to the hyperplane generated by the current and past operating points, we have that $\mathbf{U}_k \mathbf{n}_k = \mathbf{0}$ and, from (B.1) and (B.5), the hyperplane is given by $\mathbf{n}_k^\top \mathbf{u} = b_k$, with $b_k = \frac{\mathbf{n}_k \mathbf{u}_k}{\|\mathbf{n}_k\|}$. Notice that, since \mathbf{U}_k does not depend on \mathbf{u} , the direction of \mathbf{n}_k is independent of \mathbf{u} . This is not the case for the normal directions between all the other complement affine subspaces, which depend on the position of \mathbf{u} .

Upper Bound on Gradient Error

A bound on the condition number of the matrix $\mathcal{U}(\mathbf{u})$ given in (2.41) was proposed in [12, 13] in the context of a dual ISOPE approach. This bound is representative of the ill-conditioning that might occur due to the relative location of the successive operating points. However, it is not directly related to the errors resulting from truncation and measurement noise. This section introduces a coherent, although conservative, upper bound on the gradient error.

Consider the desired upper bound \mathcal{E}^U on the gradient error norm:

$$\|\boldsymbol{\epsilon}(\mathbf{u})\| = \left\| \hat{\boldsymbol{\beta}}(\mathbf{u}) - \frac{\partial \Psi}{\partial \mathbf{u}}(\mathbf{u}) \right\| \leq \|\boldsymbol{\epsilon}^t(\mathbf{u})\| + \|\boldsymbol{\epsilon}^n(\mathbf{u})\| \leq \mathcal{E}^U \quad (4.91)$$

The following theorem proposes a sufficient condition for the location of the new operating point \mathbf{u} , given the n_u operating points $\mathbf{u}_k, \mathbf{u}_{k-1}, \dots, \mathbf{u}_{k-n_u+1}$ and the upper bound \mathcal{E}^U .

Theorem 4.4 (Sufficient Condition for Gradient Accuracy)

Condition (4.91) can be satisfied by choosing \mathbf{u} such that

$$\mathcal{E}(\mathbf{u}) \leq \mathcal{E}^U, \quad (4.92)$$

where

$$\begin{aligned} \mathcal{E}(\mathbf{u}) := \mathcal{E}^t(\mathbf{u}) + \|\boldsymbol{\epsilon}^n(\mathbf{u})\|_{\max} = \frac{d_2}{2} \left\| \left[(\mathbf{u} - \mathbf{u}_k)^\top (\mathbf{u} - \mathbf{u}_k) \quad \dots \right. \right. \\ \left. \left. \dots (\mathbf{u} - \mathbf{u}_{k-n_u+1})^\top (\mathbf{u} - \mathbf{u}_{k-n_u+1}) \right] \mathcal{U}^{-1}(\mathbf{u}) \right\| + \frac{\delta}{l_{\min}(\mathbf{u})} \end{aligned} \quad (4.93)$$

Proof. The proof follows from (4.91), inequalities (4.74) and $\|\boldsymbol{\epsilon}^n\| \leq \delta/l_{\min}(\mathbf{u})$ (which follows from Proposition 4.2). \square

Remark 4.5 For given values of δ and \mathbf{d}_2 , there is a minimal value that $\mathcal{E}(\mathbf{u})$ can take. Therefore, the upper bound \mathcal{E}^U should not be selected smaller than this minimal value, otherwise the constraint (4.92) will be infeasible.

The following example analyzes the feasible regions generated by the measurement-noise component in (4.93).

Example 4.4 If $\Psi(\mathbf{u})$ is linear, $\mathbf{d}_2 = 0$ (there is no truncation error), and (4.92) and (4.93) reduce to $\delta/l_{\min}(\mathbf{u}) \leq \mathcal{E}^U$, or equivalently:

$$l_{\min}^L := \frac{\delta}{\mathcal{E}^U} \leq l_{\min}(\mathbf{u}) \quad (4.94)$$

The constraint (4.94) can be seen as the combination of the following n_b constraints:

$$l_{\min}^L = \frac{\delta}{\mathcal{E}^U} \leq l_i(\mathbf{u}), \quad i = 1, \dots, n_b \quad (4.95)$$

where n_b is the total number of complement affine subspaces. Let us consider the distance between \mathbf{u} and the hyperplane $\mathbf{n}_k^T \mathbf{u} = b_k$. If in (4.90) we have $\boldsymbol{\epsilon}^n(\mathbf{u} - \mathbf{u}_k) > 0$, the constraint (4.95) corresponding to this pair of complement affine subspaces gives

$$l_{\min}^L \leq \frac{\boldsymbol{\epsilon}^n}{\|\boldsymbol{\epsilon}^n\|}(\mathbf{u} - \mathbf{u}_k) = \frac{\mathbf{n}_k^T}{\|\mathbf{n}_k\|}(\mathbf{u} - \mathbf{u}_k),$$

which can be written as

$$\mathbf{n}_k^T \mathbf{u} \geq b_k + l_{\min}^L \|\mathbf{n}_k\|. \quad (4.96)$$

On the other hand, if $\boldsymbol{\epsilon}^n(\mathbf{u} - \mathbf{u}_k) < 0$, we obtain

$$\mathbf{n}_k^T \mathbf{u} \leq b_k - l_{\min}^L \|\mathbf{n}_k\|. \quad (4.97)$$

Hence, this point-to-hyperplane constraint generates two feasible regions, one at each side of the hyperplane $\mathbf{n}_k^T \mathbf{u} = b_k$. For all other complement affine subspaces, the direction of a vector that is normal to both affine subspaces will vary with the position of \mathbf{u} .

For the purpose of illustration, consider the two-dimensional case ($n_u = 2$) with $\delta = 0.2$ and $\mathcal{E}^U = 0.5$. Figure 4.13 uses the past operating points $\mathbf{u}_k = [0 \ -0.5]^\top$ and $\mathbf{u}_{k-1} = [0 \ 0.5]^\top$. The constraints (4.95), which can be evaluated in terms of the location of the new operating point $\mathbf{u} = [u_1 \ u_2]^\top$, consist of three point-to-line distances. For the case of the distance between \mathbf{u} and the line generated by \mathbf{u}_k and \mathbf{u}_{k-1} , denoted l_1 in this example, the feasible regions generated by (4.96) and (4.97) are given in Figure 4.13a. The feasible regions corresponding to the two remaining point-to-line distances are shown in Figures 4.13b and 4.13c. The combination of these constraints is given in Figure 4.13d. It can be seen that (4.94) generates two convex feasible regions, one on each side of the hyperplane $\mathbf{n}_k^\top \mathbf{u} = b_k$.

Example 4.5 Consider the two-dimensional case ($n_u = 2$) with $\delta = 0.2$ and $\mathbf{d}_2 = 2$. Figures 4.14a, 4.14c and 4.14e on the left use the past operating points $\mathbf{u}_k = [0 \ -0.5]^\top$ and $\mathbf{u}_{k-1} = [0 \ 0.5]^\top$, while the figures on the right, 4.14b, 4.14d and 4.14f use the past operating points $\mathbf{u}_k = [0 \ -0.1]^\top$ and $\mathbf{u}_{k-1} = [0 \ 0.1]^\top$. The upper bounds $\mathcal{E}^t(\mathbf{u})$ and $\|\epsilon^n(\mathbf{u})\|_{\max}$ can be evaluated in terms of the location of the new operating point $\mathbf{u} = [u_1 \ u_2]^\top$. Figures 4.14a and 4.14b show the contours of $\mathcal{E}^t(\mathbf{u})$. The contours corresponding to $\|\epsilon^n(\mathbf{u})\|_{\max} = \frac{\delta}{l_{\min}(\mathbf{u})}$ are shown in Figures 4.14c and 4.14d, and the contours corresponding to $\mathcal{E}(\mathbf{u})$ are shown in Figures 4.14e and 4.14f.

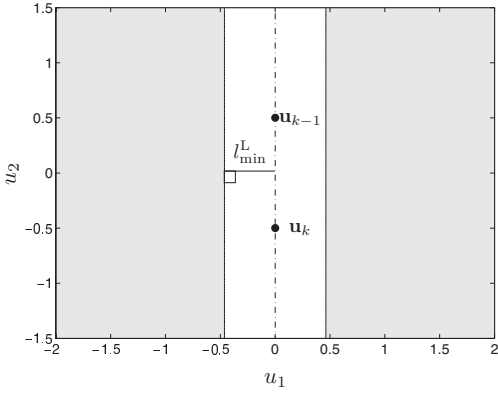
From the contours in Figure 4.14 it is seen that (i) both $\mathcal{E}^t(\mathbf{u})$ and $\|\epsilon^n(\mathbf{u})\|_{\max}$ increase when $\mathcal{U}(\mathbf{u})$ becomes ill-conditioned (\mathbf{u} aligned with \mathbf{u}_k and \mathbf{u}_{k-1}), and (ii) the two regions generated by the constraint (4.92) are nonconvex.

Convex Regions for Total Gradient Error Norm.

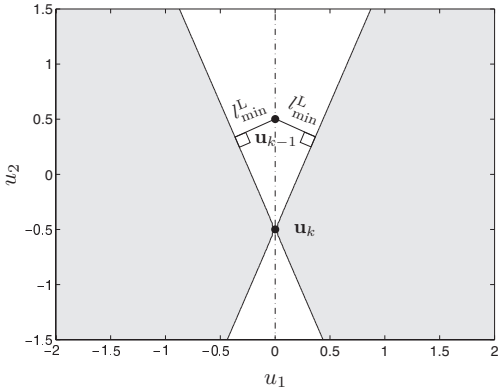
As in the ISOPE dual approach, we would like to define two optimization problems, one for each of the regions generated by the constraint (4.92). However, the fact that these regions might be non-convex creates the possibility of multiple local solutions and makes the problem more difficult to solve. In order to define convex regions, though slightly more conservative, we shall study first the geometrical properties of the upper bound on the norm of the gradient error due to truncation.

Claim 4.2 (n-Spheres) *In Proposition 4.1, $\mathcal{E}^t(\mathbf{u})$, which is defined in (4.75), gives an upper bound on the norm of the truncation error.*

a.



b.



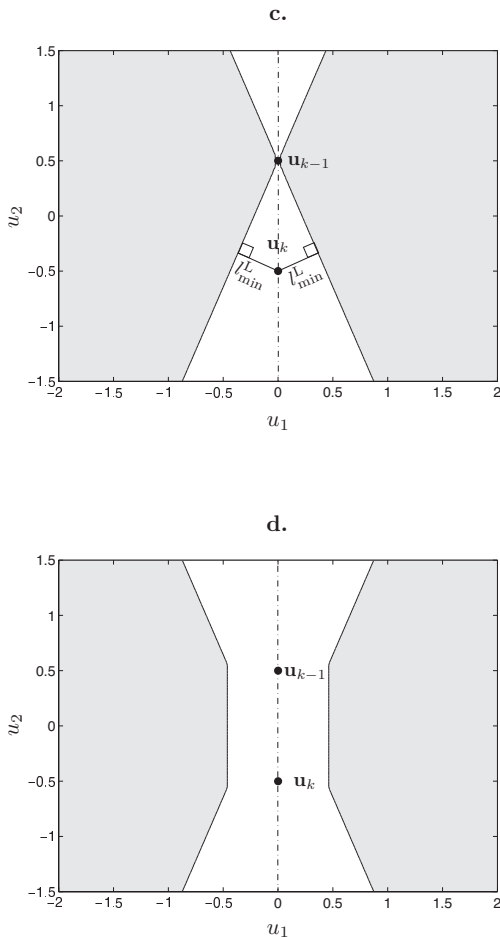


Fig. 4.13. Evaluation of the feasible regions given by constraint (4.94) in Example 4.4, as the combination of constraints (4.95). Colored area: feasible regions; **Plot a**: feasible regions given by the lower bound on the distance between $\mathbf{u} = [u_1 \ u_2]^\top$ and the line generated by \mathbf{u}_k and \mathbf{u}_{k-1} ($l_{\min}^L \leq l_1(\mathbf{u})$); **Plot b**: feasible regions given by the lower bound on the distance between \mathbf{u}_{k-1} and the line generated by \mathbf{u} and \mathbf{u}_k ($l_{\min}^L \leq l_2(\mathbf{u})$); **Plot c**: feasible regions given by the lower bound on the distance between \mathbf{u}_k and the line generated by \mathbf{u} and \mathbf{u}_{k-1} ($l_{\min}^L \leq l_3(\mathbf{u})$); **Plot d**: feasible regions given by constraint (4.94) ($l_{\min}^L \leq l_{\min}(\mathbf{u})$).

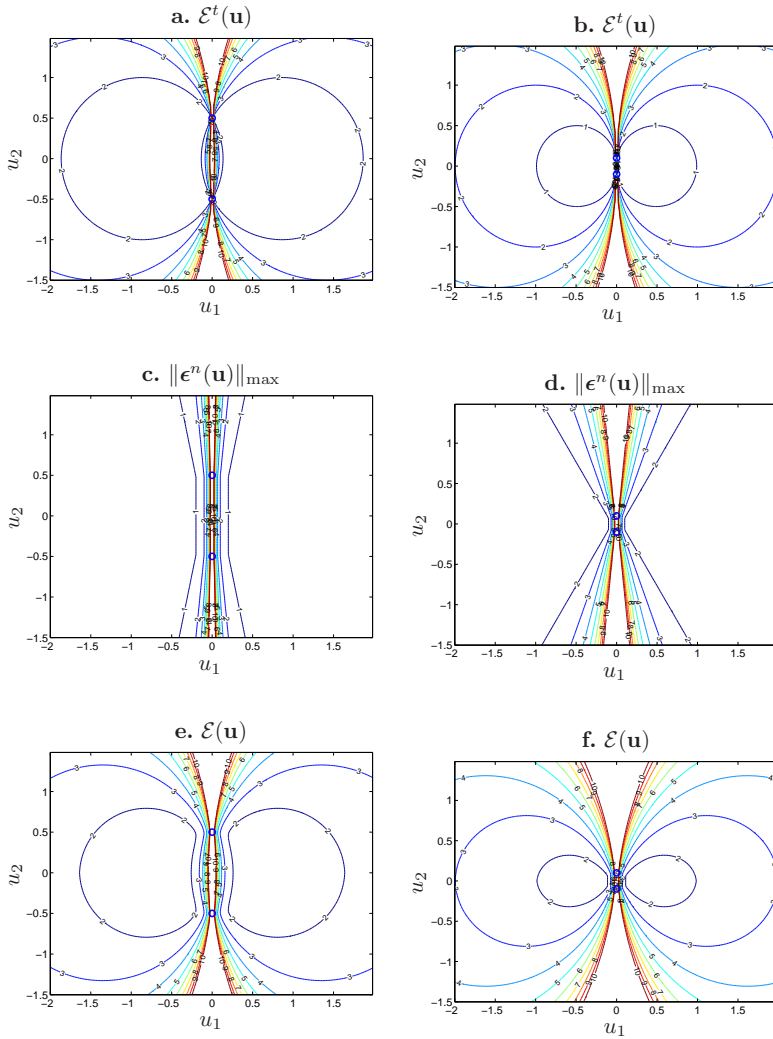


Fig. 4.14. Contour maps for an upper bound on gradient error due to truncation (a,b), measurement noise (c,d), and both (e,f) for two cases of previous points (more distant on the left and closer on the right).

For a given level c , the expression $\mathcal{E}^t(\mathbf{u}) = c$ generates two $(n_u - 1)$ -spheres of radius $r = \frac{c}{d_2}$. The centers of these $(n_u - 1)$ -spheres are symmetrically located on each side of the hyperplane generated by the n_u past operating points.

Proof. Let \mathbf{u}_c be the center of a $(n_u - 1)$ -sphere of radius r . We have:

$$r^2 = (\mathbf{u} - \mathbf{u}_c)^\top (\mathbf{u} - \mathbf{u}_c) \quad (4.98)$$

$$r^2 = (\mathbf{u}_{k-j} - \mathbf{u}_c)^\top (\mathbf{u}_{k-j} - \mathbf{u}_c), \quad j = 0, \dots, n_u - 1. \quad (4.99)$$

After expanding the right-hand sides of equations (4.98) and (4.99), extracting (4.99) from (4.98) and rearranging, we obtain:

$$\mathbf{u}^\top \mathbf{u} - \mathbf{u}_{k-j}^\top \mathbf{u}_{k-j} = 2 \mathbf{u}_c^\top (\mathbf{u} - \mathbf{u}_{k-j}), \quad j = 0, \dots, n_u - 1. \quad (4.100)$$

Equations (4.100) can be written in matrix form as follows:

$$[\mathbf{u}^\top \mathbf{u} - \mathbf{u}_k^\top \mathbf{u}_k \quad \dots \quad \mathbf{u}^\top \mathbf{u} - \mathbf{u}_{k-n_u+1}^\top \mathbf{u}_{k-n_u+1}] = 2\mathbf{u}_c^\top \mathcal{U}(\mathbf{u}) \quad (4.101)$$

From (4.101) and the following expansions:

$$\begin{aligned} (\mathbf{u} - \mathbf{u}_{k-j})^\top (\mathbf{u} - \mathbf{u}_{k-j}) &= \mathbf{u}^\top \mathbf{u} - \mathbf{u}_{k-j}^\top \mathbf{u}_{k-j} - 2 \mathbf{u}^\top \mathbf{u}_{k-j} + 2\mathbf{u}_{k-j}^\top \mathbf{u}_{k-j}, \\ j &= 0, \dots, n_u - 1, \end{aligned}$$

we can write

$$\begin{aligned} [(\mathbf{u} - \mathbf{u}_k)^\top (\mathbf{u} - \mathbf{u}_k) \quad \dots \quad (\mathbf{u} - \mathbf{u}_{k-n_u+1})^\top (\mathbf{u} - \mathbf{u}_{k-n_u+1})] \mathcal{U}^{-1}(\mathbf{u}) &= \\ = 2\mathbf{u}_c^\top - 2[\mathbf{u}_k^\top (\mathbf{u} - \mathbf{u}_k) \quad \dots \quad \mathbf{u}_{k-n_u+1}^\top (\mathbf{u} - \mathbf{u}_{k-n_u+1})] \mathcal{U}^{-1}(\mathbf{u}) \end{aligned} \quad (4.102)$$

Noticing that

$$\begin{aligned} [(\mathbf{u} - \mathbf{u}_k)^\top (\mathbf{u} - \mathbf{u}_k) \quad \dots \quad (\mathbf{u} - \mathbf{u}_{k-n_u+1})^\top (\mathbf{u} - \mathbf{u}_{k-n_u+1})] \mathcal{U}^{-1}(\mathbf{u}) \\ + [\mathbf{u}_k^\top (\mathbf{u} - \mathbf{u}_k) \quad \dots \quad \mathbf{u}_{k-n_u+1}^\top (\mathbf{u} - \mathbf{u}_{k-n_u+1})] \mathcal{U}^{-1}(\mathbf{u}) = \mathbf{u}^\top, \end{aligned}$$

(4.102) reduces to:

$$\begin{aligned} [(\mathbf{u} - \mathbf{u}_k)^\top (\mathbf{u} - \mathbf{u}_k) \quad \dots \quad (\mathbf{u} - \mathbf{u}_{k-n_u+1})^\top (\mathbf{u} - \mathbf{u}_{k-n_u+1})] \mathcal{U}^{-1}(\mathbf{u}) \\ = 2(\mathbf{u}^\top - \mathbf{u}_c^\top) \end{aligned}$$

Therefore we can write,

$$c = \frac{d_2}{2} \left\| \left[(\mathbf{u} - \mathbf{u}_k)^\top (\mathbf{u} - \mathbf{u}_k) \dots \right. \right. \quad (4.103)$$

$$\left. \left. \dots (\mathbf{u} - \mathbf{u}_{k-n_u+1})^\top (\mathbf{u} - \mathbf{u}_{k-n_u+1}) \right] \mathcal{U}^{-1}(\mathbf{u}) \right\| = d_2 \|\mathbf{u} - \mathbf{u}_c\|$$

which corresponds to the equation of a $(n_u - 1)$ -sphere of radius $r = \frac{c}{d_2}$.

The equations (4.99) being quadratic, for a given value of r , they provide two possible solutions \mathbf{u}_{c1} and \mathbf{u}_{c2} for the center point. It is obvious from Euclidean geometry that these center points are located symmetrically with respect to the hyperplane $\mathbf{n}_k^\top \mathbf{u} = b_k$. \square

The situation in the two-dimensional case is represented in Figure 4.15. The current and past operating points are \mathbf{u}_k and \mathbf{u}_{k-1} . The center points $\mathbf{u}_{c1}(\mathbf{u})$ and $\mathbf{u}_{c2}(\mathbf{u})$ as well as the radius $r(\mathbf{u})$ vary with the position of the new operating point \mathbf{u} . The corresponding upper bound on the deterministic error norm is $\mathcal{E}^t(\mathbf{u}) = d_2 r(\mathbf{u})$.

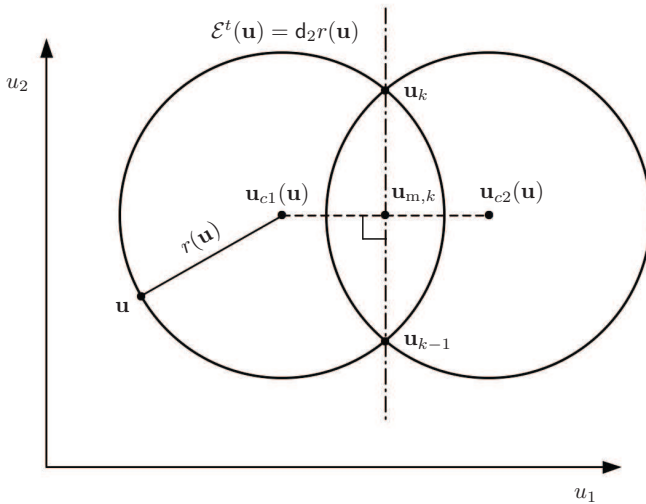


Fig. 4.15. Representation of the geometrical properties of $\mathcal{E}^t(\mathbf{u})$ in the two-dimensional case.

From (4.101) it follows that for a given operating point \mathbf{u} , the center point is given by

$$\mathbf{u}_c^\top(\mathbf{u}) = \frac{1}{2} \left[\mathbf{u}^\top \mathbf{u} - \mathbf{u}_k^\top \mathbf{u}_k \cdots \mathbf{u}^\top \mathbf{u} - \mathbf{u}_{k-n_u+1}^\top \mathbf{u}_{k-n_u+1} \right] \mathcal{U}^{-1}(\mathbf{u}) \quad (4.104)$$

The worst-case measurement error $\delta/l_{\min}(\mathbf{u})$ being convex on each side of the hyperplane $\mathbf{n}_k^\top \mathbf{u} = b_k$ (see Example 4.4), it can be concluded that the non-convexity of the regions generated by the constraint (4.92) is due to the part of the $(n_u - 1)$ -spheres that crosses the hyperplane $\mathbf{n}_k^\top \mathbf{u} = b_k$. The distance (positive or negative) from the center point $\mathbf{u}_c(\mathbf{u})$ to the hyperplane $\mathbf{n}_k^\top \mathbf{u} = b_k$ is given by:

$$l_C(\mathbf{u}) = \frac{b_k - \mathbf{n}_k^\top \mathbf{u}_c(\mathbf{u})}{\|\mathbf{n}_k\|}. \quad (4.105)$$

Given the n_u operating points $\mathbf{u}_k, \mathbf{u}_{k-1}, \dots, \mathbf{u}_{k-n_u+1}$, the point $\mathbf{u}_{m,k}$ can be obtained by projecting the center point $\mathbf{u}_c(\mathbf{u})$ on the hyperplane $\mathbf{n}_k^\top \mathbf{u} = b_k$:

$$\mathbf{u}_{m,k} = \mathbf{u}_c(\mathbf{u}) + \frac{l_C(\mathbf{u})}{\|\mathbf{n}_k\|} \mathbf{n}_k \quad (4.106)$$

It can be verified that $\mathbf{u}_{m,k}$ is independent of \mathbf{u} (see the location of $\mathbf{u}_{m,k}$ in Figure 4.15 for the two-dimensional case). For a given upper bound \mathcal{E}^U , it is possible to define convex feasibility regions by adding constraints expressing the distance between the new point and the hyperplane, which eliminates the non-convex part of the regions generated by (4.92), as illustrated in Figure 4.16. The minimal point-to-hyperplane distance ρ_k can be determined numerically by finding the smallest absolute value solution to the following equation:

$$\mathcal{E} \left(\mathbf{u}_{m,k} + \frac{\rho_k}{\|\mathbf{n}_k\|} \mathbf{n}_k \right) = \mathcal{E}^U$$

4.2.2 Dual Modifier Adaptation for Unconstrained Optimization

In the unconstrained optimization case, the modifier-adaptation approach requires an estimate of the plant cost gradient to be available at each iteration. The function $\Psi(\mathbf{u})$ introduced in (4.66) is taken as the cost function, i.e. $\Psi(\mathbf{u}) = \Phi_p(\mathbf{u})$. In particular, it is assumed that the cost function can be evaluated from the noisy output measurements as follows:

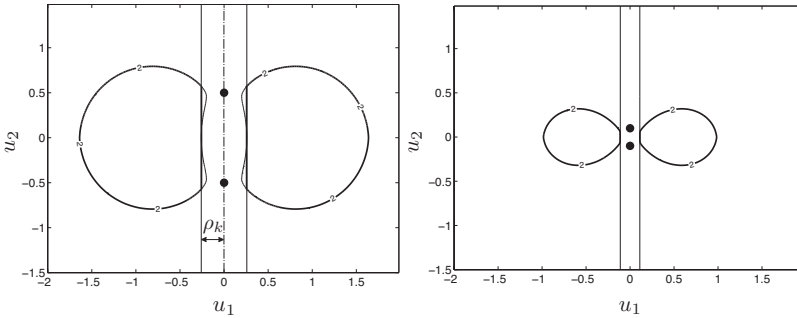


Fig. 4.16. Convex regions (in bold) for locating the operating point \mathbf{u} in Example 4.5 with $\mathcal{E}(\mathbf{u}) \leq 2$.

$$\psi(\mathbf{u}) = \phi(\mathbf{u}, \mathbf{y}_p(\mathbf{u}) + \boldsymbol{\nu}) = \overline{\Phi}_p(\mathbf{u}) + v \quad (4.107)$$

where $\boldsymbol{\nu}$ is the output measurement noise vector and v represents the resulting noise in the cost function. Notice that, even if $\boldsymbol{\nu}$ has zero mean, v might have a nonzero mean if the function $\phi(\mathbf{u}, \mathbf{y})$ is nonlinear in \mathbf{y} .

Error in Cost Function Gradient

This subsection analyzes how the error in the estimated gradient of the cost function affects the ability to ensure that the estimated gradient provides a correct descent direction. The following lemma provides a sufficient condition for having a true descent direction.

Lemma 4.1 (Sufficient Condition for a Local Descent Direction) *If at a given operating point \mathbf{u} the gradient error norm $\|\boldsymbol{\epsilon}(\mathbf{u})\|$ is smaller than the true gradient norm $\|\frac{\partial \Phi_p}{\partial \mathbf{u}}(\mathbf{u})\|$, then the direction $-\hat{\boldsymbol{\beta}}(\mathbf{u})$ provided by the gradient estimator is a descent direction for the cost Φ_p at \mathbf{u} .*

Proof. For the gradient estimator $\hat{\boldsymbol{\beta}}(\mathbf{u}) = \frac{\partial \overline{\Phi}_p}{\partial \mathbf{u}}(\mathbf{u}) + \boldsymbol{\epsilon}(\mathbf{u})$, one can write:

$$\begin{aligned} -\frac{\partial \overline{\Phi}_p}{\partial \mathbf{u}}(\mathbf{u})\hat{\boldsymbol{\beta}}(\mathbf{u})^\top &= -\left\| \frac{\partial \overline{\Phi}_p}{\partial \mathbf{u}}(\mathbf{u}) \right\|^2 - \frac{\partial \overline{\Phi}_p}{\partial \mathbf{u}}(\mathbf{u})\boldsymbol{\epsilon}(\mathbf{u})^\top = \quad (4.108) \\ &= -\left\| \frac{\partial \overline{\Phi}_p}{\partial \mathbf{u}}(\mathbf{u}) \right\|^2 - \left\| \frac{\partial \overline{\Phi}_p}{\partial \mathbf{u}}(\mathbf{u}) \right\| \|\boldsymbol{\epsilon}(\mathbf{u})\| \cos \theta, \end{aligned}$$

where θ is the angle between $\frac{\partial\Phi_p}{\partial\mathbf{u}}(\mathbf{u})$ and $\boldsymbol{\epsilon}(\mathbf{u})$. From (4.108), if $\|\boldsymbol{\epsilon}(\mathbf{u})\| < \left\|\frac{\partial\Phi_p}{\partial\mathbf{u}}(\mathbf{u})\right\|$, then $-\frac{\partial\Phi_p}{\partial\mathbf{u}}(\mathbf{u})\hat{\boldsymbol{\beta}}(\mathbf{u})^\top < 0$ for any θ , and $-\hat{\boldsymbol{\beta}}(\mathbf{u})$ is a descent direction for Φ_p at \mathbf{u} . \square

The following theorem gives sufficient conditions for a descent direction to be found at any operating point \mathbf{u} located outside a ball of a certain radius centered at the plant optimum.

Theorem 4.5 (Sufficient Condition for Correct Descent Direction outside a Ball) *Assume $\Phi_p(\mathbf{u})$ is a convex function twice continuously differentiable with respect to \mathbf{u} , for which \mathbf{u}_p^* is a strict local minimum. Let d_2^L be the smallest absolute eigenvalue of the Hessian of $\Phi_p(\mathbf{u})$, which is assumed to be different from zero. If the gradient error norm $\|\boldsymbol{\epsilon}(\mathbf{u})\|$ satisfies $\|\boldsymbol{\epsilon}(\mathbf{u})\| \leq \mathcal{E}^U$, then, for each \mathbf{u} with $\|\mathbf{u} - \mathbf{u}_p^*\| > \frac{\mathcal{E}^U}{d_2^L}$, the vector $-\hat{\boldsymbol{\beta}}(\mathbf{u})$ is a descent direction for Φ_p at \mathbf{u} .*

Proof. Using Taylor's formula around \mathbf{u}_p^* , and noticing that $\frac{\partial\Phi_p}{\partial\mathbf{u}}(\mathbf{u}_p^*) = \mathbf{0}$, we can write:

$$\frac{\partial\Phi_p}{\partial\mathbf{u}}(\mathbf{u}) = (\mathbf{u} - \mathbf{u}_p^*)^\top \frac{\partial^2\Phi_p}{\partial\mathbf{u}^2}(\mathbf{u}_p^* + \vartheta(\mathbf{u} - \mathbf{u}_p^*)), \quad (4.109)$$

where, according to the mean value theorem, $\exists \vartheta \in [0, 1]$ such that (4.109) holds. Since $\Phi_p(\mathbf{u})$ is a twice continuously differentiable convex function, the Hessian $\frac{\partial^2\Phi_p}{\partial\mathbf{u}^2}(\mathbf{u}_p^* + \vartheta(\mathbf{u} - \mathbf{u}_p^*))$ must be positive semi-definite. Since we also assume that $d_2^L \neq 0$, then $\frac{\partial^2\Phi_p}{\partial\mathbf{u}^2}(\mathbf{u}_p^* + \vartheta(\mathbf{u} - \mathbf{u}_p^*))$ must be positive definite. Let us define the auxiliary gradient $\frac{\partial\Phi_p^L}{\partial\mathbf{u}}(\mathbf{u}) := d_2^L(\mathbf{u} - \mathbf{u}_p^*)^\top$. Since the Hessian matrix $\frac{\partial^2\Phi_p}{\partial\mathbf{u}^2}(\mathbf{u})$ is positive definite for any \mathbf{u} , and d_2^L is the smallest absolute value of the second-order derivatives of $\Phi_p(\mathbf{u})$, we can write:

$$d_2^L\|\mathbf{u} - \mathbf{u}_p^*\| \leq \left\|\frac{\partial\Phi_p}{\partial\mathbf{u}}(\mathbf{u})\right\| \quad (4.110)$$

Then, for $\|\mathbf{u} - \mathbf{u}_p^*\| > \frac{\mathcal{E}^U}{d_2^L}$, we have:

$$\|\boldsymbol{\epsilon}(\mathbf{u})\| \leq \mathcal{E}^U < d_2^L\|\mathbf{u} - \mathbf{u}_p^*\| \leq \left\|\frac{\partial\Phi_p}{\partial\mathbf{u}}(\mathbf{u})\right\| \quad (4.111)$$

From (4.111), $\|\boldsymbol{\epsilon}(\mathbf{u})\| < \left\|\frac{\partial\Phi_p}{\partial\mathbf{u}}(\mathbf{u})\right\|$, and therefore $-\hat{\boldsymbol{\beta}}(\mathbf{u})$ is a descent direction for Φ_p from Lemma 4.1. \square

The result of Theorem 4.5 is illustrated in Figure 4.17. If $\|\epsilon(\mathbf{u})\| \leq \mathcal{E}^U$, then for any r such that $r > \frac{\mathcal{E}^U}{d_2^L}$, we will have $\|\epsilon(\mathbf{u})\| < \|\frac{\partial \Phi_p}{\partial \mathbf{u}}(\mathbf{u})\|$.

Theorem 4.5 provides a sufficient condition for $-\hat{\beta}(\mathbf{u})$ to be a descent direction for Φ_p at \mathbf{u} . However, the descent direction $-\hat{\beta}(\mathbf{u})$ may point outside the ball of radius $\|\mathbf{u} - \mathbf{u}_p^*\|$ centered at \mathbf{u}_p^* . In practice, the value of d_2^L can be zero or close to zero, in which case the sufficient conditions of Theorem 4.5 are very conservative and, therefore, of little practical value since it is only proven that $-\hat{\beta}(\mathbf{u})$ is a descent direction for \mathbf{u} chosen outside a ball of very large radius.

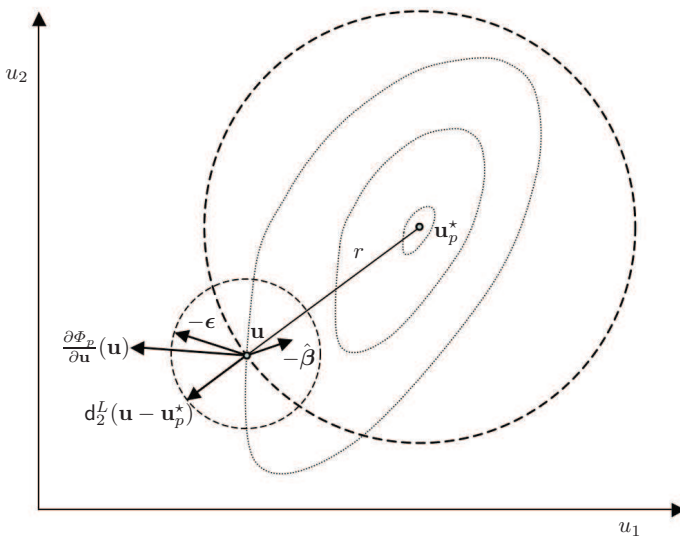


Fig. 4.17. Illustration of the result of Theorem 4.5. Dotted curves: contours of $\Phi_p(\mathbf{u})$; Big dashed circle: Ball of radius r centered at \mathbf{u}_p^* ; Small dashed circle: Ball of radius $d_2^L r$ centered at \mathbf{u} .

Dual Modifier-Adaptation Algorithm

The dual modifier-adaptation scheme proposed in this section uses the upper bound on the gradient error defined in Subsection 4.2.1 as a constraint in the optimization problem. On each side of the hyperplane

generated by the current and past operating points, $\mathbf{n}_k^\top \mathbf{u} = b_k$, a modified model-based optimization problem is solved. The optimization problem corresponding to the half space $\mathbf{n}_k^\top \mathbf{u} \geq b_k$ reads:

$$\begin{aligned} \mathbf{u}_{k+1}^+ &= \arg \min_{\mathbf{u}} \Phi_m(\mathbf{u}, \boldsymbol{\theta}) = \Phi(\mathbf{u}, \boldsymbol{\theta}) + \boldsymbol{\lambda}_k^{\Phi^\top} \mathbf{u} & (4.112) \\ \text{s.t. } \mathcal{E}(\mathbf{u}) &\leq \mathcal{E}^U \\ \mathbf{n}_k^\top \mathbf{u} &\geq b_k + \rho_k \|\mathbf{n}_k\| \end{aligned}$$

while, for the half space $\mathbf{n}_k^\top \mathbf{u} \leq b_k$, one has:

$$\begin{aligned} \mathbf{u}_{k+1}^- &= \arg \min_{\mathbf{u}} \Phi_m(\mathbf{u}, \boldsymbol{\theta}) = \Phi(\mathbf{u}, \boldsymbol{\theta}) + \boldsymbol{\lambda}_k^{\Phi^\top} \mathbf{u} & (4.113) \\ \text{s.t. } \mathcal{E}(\mathbf{u}) &\leq \mathcal{E}^U \\ \mathbf{n}_k^\top \mathbf{u} &\leq b_k - \rho_k \|\mathbf{n}_k\| \end{aligned}$$

The modifiers $\boldsymbol{\lambda}_k^{\Phi^\top}$ are adapted as in (4.13). The next operating point is chosen as the value of $\{\mathbf{u}_{k+1}^+, \mathbf{u}_{k+1}^-\}$ that minimizes the augmented cost function $\Phi_m(\mathbf{u}, \boldsymbol{\theta})$.

Initialization.

In order to initialize the dual modifier-adaptation scheme, it is necessary to count initially with $(n_u + 1)$ operating points satisfying the constraint $\mathcal{E}(\mathbf{u}) \leq \mathcal{E}^U$ in order to obtain a first estimate of the gradient and to evaluate the gradient modifiers for the first time. One possibility is to obtain the initial $(n_u + 1)$ operating points by making deviations from the initial point along the Cartesian axes, as in the FFD scheme (2.37). This technique was proposed for the dual ISOPE algorithm [12, 13], and is retained here. Notice that an optimized initial phase has also been proposed for the dual ISOPE algorithm [13]. How a similar approach could be implemented when considering an upper bound on the gradient error is a possible subject of future research.

4.2.3 Dual Modifier Adaptation for Constrained Optimization

In the case of a constrained optimization problem, the modifier-adaptation approach requires an estimate of the cost and constraint gradients for the plant to be available at each iteration. In order to implement dual modifier adaptation with a bound on the gradient error,

an extra question that arises is which gradient error should be considered. The cost and constraint functions will be evaluated from the noisy measurements with different resulting noise levels, and the curvature of these functions will be different as well. The implementation of the upper bound on the gradient error introduced in Subsection 4.2.1 requires the selection of the parameters δ and \mathbf{d}_2 corresponding to the function of interest $\Psi(\mathbf{u})$, presented in (4.66). In the case of an unconstrained optimization problem, this function was selected as the cost function. In the case of constrained optimization problems, two possible strategies are:

Strategy 1.

Determine values of δ and \mathbf{d}_2 for the cost function and each of the constraints individually, and select the highest value of δ and the highest value of \mathbf{d}_2 . This guarantees that the upper bound on the gradient error given by $\mathcal{E}(\mathbf{u})$ is valid for the cost function and for each constraint individually. However, this strategy will introduce additional conservatism, leading to an implementation with smaller regions for placing the new operating point.

Strategy 2.

Select the function $\Psi(\mathbf{u})$ as a linear combination of the cost and constraint functions. That is:

$$\begin{aligned}\psi(\mathbf{u}) &= \phi(\mathbf{u}, \mathbf{y}_p(\mathbf{u}) + \boldsymbol{\nu}) + \mathbf{c}^T \mathbf{g}(\mathbf{u}, \mathbf{y}_p(\mathbf{u}) + \boldsymbol{\nu}) \\ &= \bar{\Phi}_p(\mathbf{u}) + \mathbf{c}^T \mathbf{G}_p(\mathbf{u}) + v\end{aligned}\quad (4.114)$$

where $\boldsymbol{\nu}$ is the output measurement noise and v represents the resulting noise in $\Psi(\mathbf{u})$. Notice that, even if $\boldsymbol{\nu}$ is zero mean, v might have a nonzero mean if any of the functions $\phi(\mathbf{u}, \mathbf{y})$ and $g_i(\mathbf{u}, \mathbf{y})$, $i = 1, \dots, n_g$, is nonlinear in \mathbf{y} .

In dual modifier adaptation, we are interested in keeping track of the infimum value of the plant cost function. To limit the error of the individual constraint gradients is not an objective per se; the effect that these errors will have on the offset with respect to the infimum of the plant is of prime importance. Hence, one natural choice is to select the weights on the constraint functions as the Lagrange multipliers, that is, $\mathbf{c} = \boldsymbol{\mu}$. In this case, the function $\Psi(\mathbf{u})$ corresponds to the Lagrangian function. Since the Lagrange multipliers associated

with inactive constraints are equal to zero, this choice eliminates from $\Psi(\mathbf{u})$ the inactive constraints. It is clear that, if a constraint does not become active in modifier adaptation, the error with which its gradient is estimated will not influence the behavior of the algorithm. However, the difficulty with this choice is that the active constraints are not assumed to be known a priori. One possibility is to update the values of \mathbf{c} as $\mathbf{c}_k = \boldsymbol{\mu}_k$. Notice that this will modify the level of noise v at each RTO iteration. Therefore, the values of δ and \mathbf{d}_2 could also be updated in the upper bound $\mathcal{E}(\mathbf{u})$ at each RTO iteration. The difficulty in doing so is that, if δ_k and $\mathbf{d}_{2,k}$ are varied, there is no guarantee that the constraint (4.92) will remain feasible since the past operating points were placed using different values of δ_k and $\mathbf{d}_{2,k}$. In general, infeasibilities can occur if the value of δ_k or $\mathbf{d}_{2,k}$ decreases from one RTO period to the other. The way to deal with these infeasibilities constitutes a subject of future research. If the problem becomes infeasible, it is possible to (i) reinitialize the algorithm with a FFD gradient estimation, which requires generating n_u new operating points, or (ii) increase the value of the upper bound \mathcal{E}^U . A methodology for implementing these approaches is not part of this thesis.

Dual Modifier-Adaptation Algorithm

Once the parameters δ and \mathbf{d}_2 are selected, the following modified model-based optimization problems including constraints are solved on each side of the hyperplane generated by the n_u past operating points, $\mathbf{n}_k^T \mathbf{u} = b_k$. The optimization problem corresponding to the half space $\mathbf{n}_k^T \mathbf{u} \geq b_k$ reads:

$$\begin{aligned} \mathbf{u}_{k+1}^+ &= \arg \min_{\mathbf{u}} \Phi_m(\mathbf{u}, \boldsymbol{\theta}) = \Phi(\mathbf{u}, \boldsymbol{\theta}) + \boldsymbol{\lambda}_k^{\Phi T} \mathbf{u} & (4.115) \\ \text{s.t. } \quad & \mathbf{G}_m(\mathbf{u}, \boldsymbol{\theta}) = \mathbf{G}(\mathbf{u}, \boldsymbol{\theta}) + \boldsymbol{\varepsilon}_k + \boldsymbol{\lambda}_k^{\mathbf{G} T} (\mathbf{u} - \mathbf{u}_k) \leq \mathbf{0} \\ & \mathcal{E}(\mathbf{u}) \leq \mathcal{E}^U \\ & \mathbf{n}_k^T \mathbf{u} \geq b_k + \rho_k \|\mathbf{n}_k\| \\ & \mathbf{u}^L \leq \mathbf{u} \leq \mathbf{u}^U. \end{aligned}$$

while, the for the half space $\mathbf{n}_k^T \mathbf{u} \leq b_k$, one has:

$$\mathbf{u}_{k+1}^- = \arg \min_{\mathbf{u}} \Phi_m(\mathbf{u}, \boldsymbol{\theta}) = \bar{\Phi}(\mathbf{u}, \boldsymbol{\theta}) + \boldsymbol{\lambda}_k^{\phi \top} \mathbf{u} \quad (4.116)$$

$$\begin{aligned} \text{s.t. } \quad & \mathbf{G}_m(\mathbf{u}, \boldsymbol{\theta}) = \mathbf{G}(\mathbf{u}, \boldsymbol{\theta}) + \varepsilon_k + \boldsymbol{\lambda}_k^{\mathbf{G} \top} (\mathbf{u} - \mathbf{u}_k) \leq \mathbf{0} \\ & \mathcal{E}(\mathbf{u}) \leq \mathcal{E}^{\text{U}} \\ & \mathbf{n}_k^{\top} \mathbf{u} \leq b_k - \rho_k \|\mathbf{n}_k\| \\ & \mathbf{u}^{\text{L}} \leq \mathbf{u} \leq \mathbf{u}^{\text{U}}. \end{aligned}$$

The modifiers ε_k , $\boldsymbol{\lambda}_k^{\mathbf{G}}$ and $\boldsymbol{\lambda}_k^{\phi}$ are adapted as in (4.10). The next operating point is chosen as the value of $\{\mathbf{u}_{k+1}^+, \mathbf{u}_{k+1}^-\}$ that minimizes the augmented cost function $\Phi_m(\mathbf{u}, \boldsymbol{\theta})$.

It might happen that one of the optimization problems (4.115) or (4.116) is infeasible. However, if the plant $\mathbf{y}_p(\mathbf{u})$ is assumed to be unchanged during the implementation, it will not happen that both problems (4.115) and (4.116) become infeasible at the same time, since when facing constraints the adaptation can always return from the same way it came. That is, given the current and past operating points $\mathbf{u}_k, \mathbf{u}_{k-1}, \dots, \mathbf{u}_{k-n_u+1}$, a new operating point \mathbf{u}_{k+1} satisfying $\mathcal{E}(\mathbf{u}_{k+1}) \leq \mathcal{E}^{\text{U}}$ can always be found as $\mathbf{u}_{k+1} = \mathbf{u}_{k-n_u}$.

Also, since most numerical optimization solvers require a feasible initial guess, in order to apply the approach automatically, a procedure is required in order to find a feasible initial guess prior to each optimization and to deal with infeasibilities of problems (4.115) or (4.116). This infeasibility issue, which is also present in the dual ISOPE approach, has not been addressed in the ISOPE literature.

In summary, when constraints are included in the optimization problem, two main issues arise in the implementation of dual modifier adaptation with a bound on the gradient error. The first issue is the strategy used in order to select the parameters δ and \mathbf{d}_2 . The second issue is how to deal with possible infeasibilities in optimization problems (4.115) and (4.116).

4.2.4 Case Study: Williams-Otto Reactor

The dual modifier-adaptation approach is applied to the Williams-Otto reactor example described in Subsection 2.2.

The inputs are scaled using the intervals [3, 6] for F_B , and [70, 100] for T_R . In this range, the maximal value of \mathbf{d}_2 obtained with the scaled inputs is $\mathbf{d}_2 = 1030$ for the model (the unknown actual plant value is $\mathbf{d}_2 = 1221$). The simulations are carried out assuming that the cost

function noise v has a Gaussian zero-mean distribution with standard deviation $\sigma_\phi = 0.5$. The noise interval $\delta = 3$ is chosen.

Modifier Adaptation with FFD Approach.

First, the FFD approach is applied, which consists in perturbing each input individually around the current operating point with a step size h . In the two-dimensional case, the shortest distance between complement affine subspaces for the FFD arrangement is $l^{\min} = \frac{1}{\sqrt{2}}h$. Hence, from (4.79) and (4.93) we have

$$\mathcal{E}(h) = \frac{\sqrt{2}}{2} h \mathbf{d}_2 + \sqrt{2} \frac{\delta}{h} \quad (4.117)$$

The step size that minimizes $\mathcal{E}(h)$ is $h^* = \sqrt{\frac{2\delta}{\mathbf{d}_2}} = 0.0763$ (scaled value), for which $\mathcal{E}(h^*) = 111.2$. The approach is applied without including noise (Figure 4.18). In Figures 4.18a, 4.18b, 4.18c and 4.18d, the modifier-adaptation scheme is applied estimating the cost function gradient using the FFD approach with $h = h^*$. The exponential filter (4.13) is used with gain parameter $d = 0.5$ in Figure 4.18a and $d = 1$ in Figure 4.18b. The observed offset with respect to the true optimum is due to the gradient error due to truncation. The direction of the finite-difference perturbations has an influence on the gradient error due to truncation, which is made evident by the observed difference in the converged solutions in Figures 4.18c and 4.18d. If the step size h is decreased, the iterations converge closer to the plant optimum as observed in Figures 4.18e and 4.18f.

Next, the approach is applied by adding measurement noise. For 10 noise realizations, the input trajectories corresponding to 15 RTO iterations are presented in Figure 4.19. Since, at each RTO iteration, two perturbations are required to estimate the gradient, the total number of operating points is 45. The figures in the left, 4.19a, 4.19c and 4.19e show the input trajectories, and the figures in the right, 4.19b, 4.19d and 4.19f, show the corresponding evolution of the cost. The true value of the plant profit is plotted, and not the noisy one. The step size is $h = 0.12$ in Figures 4.19a and 4.19b, $h = 0.0763$ in Figures 4.19c and 4.19d, and $h = 0.02$ in Figures 4.19e and 4.19f. Notice that the optimal step size evaluated from (4.117) gives a good compromise between loss of profit due to truncation error and measurement error.

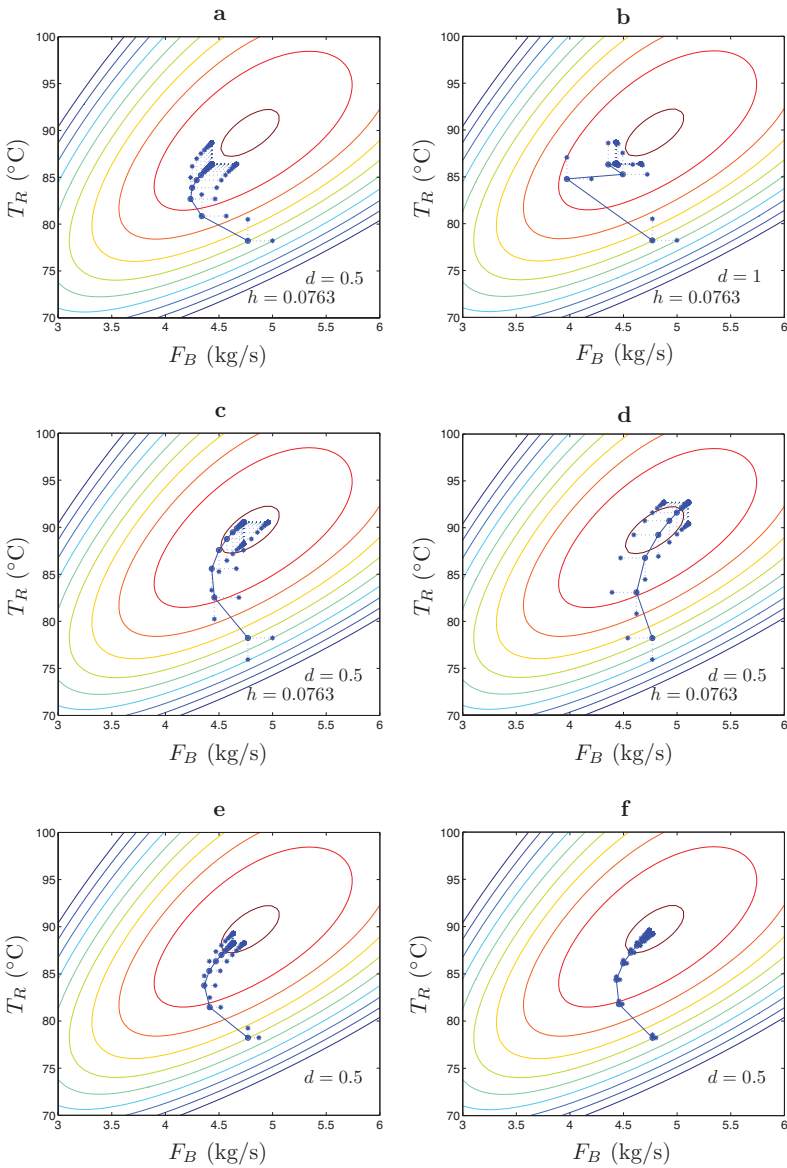


Fig. 4.18. Input trajectory for modifier adaptation with FFD. Simulations without noise.

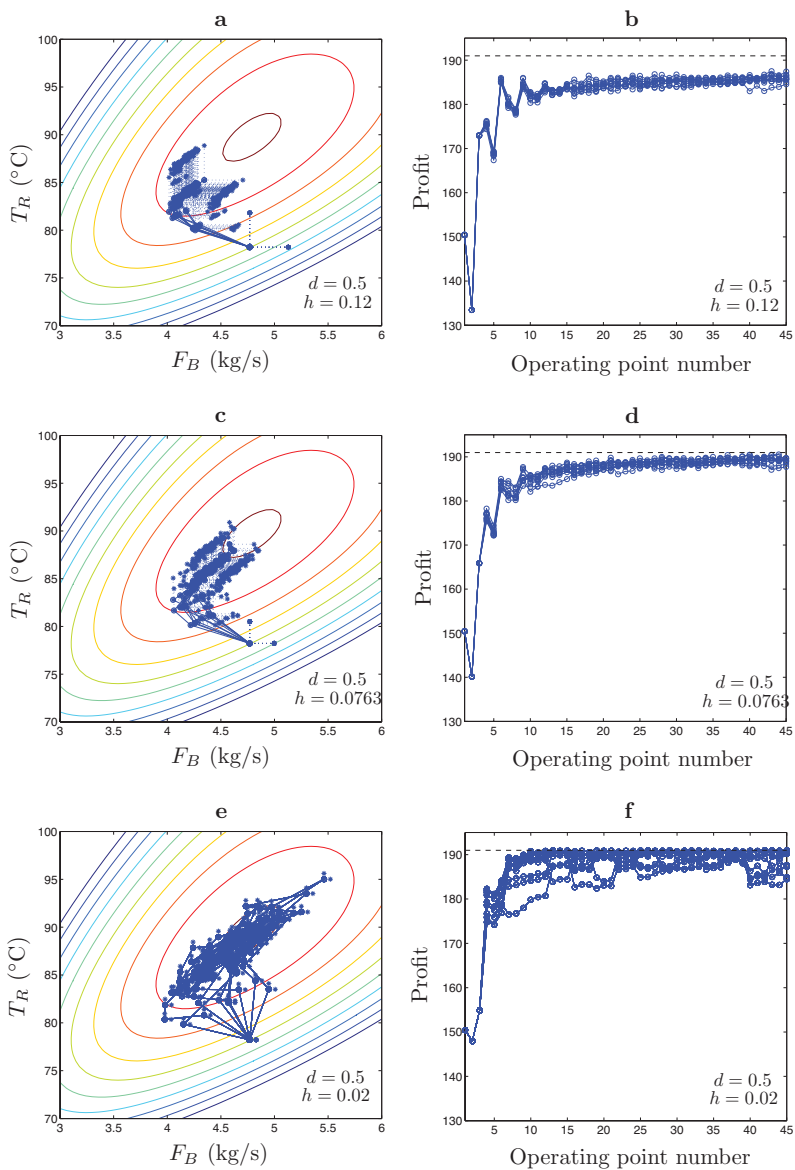


Fig. 4.19. Input and cost trajectories for modifier adaptation with FFD. Simulations with noise.

Dual Modifier Adaptation with Bound on Gradient Error.

Dual modifier adaptation is applied with $\mathcal{E}^U = 111.2$. The algorithm is initialized using FFD with $h = 0.0763$. The resulting input trajectory without noise is shown in Figure 4.20a with the exponential filter (4.13) implemented with $d = 0.5$, and in Figure 4.20b with $d = 1$. These figures can be compared with Figures 4.18a and 4.18b. Next, the approach is applied by adding measurement noise. For 10 noise realizations, the input trajectories corresponding to 45 RTO iterations are presented in Figure 4.21. It can be seen that, for this particular case study, dual modifier adaptation shows a faster convergence to the neighborhood of the plant optimum than modifier adaptation with the FFD approach (compare Figures 4.21b and 4.19d). The offset in the converged solution experienced with FFD is not observed with dual modifier adaptation.

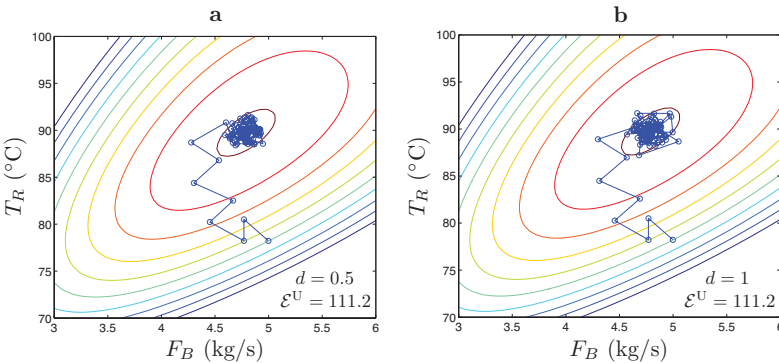


Fig. 4.20. Input trajectory with dual modifier adaptation. Simulations without noise.

Figure 4.22a shows the evolution of the plant profit and the gradient error norm for 20 noise realizations. At the operating point 20, the flow rate F_A is increased from 1.8275 kg/s to 2.2 kg/s. In the present implementation, this flow rate change is assumed to be known, and the model is updated with the new value. Modifier adaptation tracks the change in the plant optimum. It can be seen in the upper plot of Figure 4.22a that, within 6 iterations, the neighborhood of the new optimum profit is reached for all 20 realizations. Also, the lower plot of Figure 4.22a shows that the gradient error norm is kept lower than

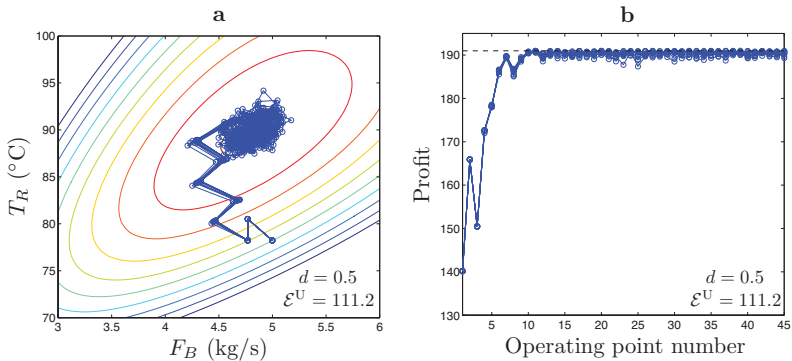


Fig. 4.21. Input and cost trajectories with dual modifier adaptation. Simulations with noise.

\mathcal{E}^U . The observed peak in gradient error occurring at the operating points 21 and 22 is due to the fact that, at these points, the gradient is estimated using operating points with different values of F_A .

Dual Modifier Adaptation with Bound on Condition Number.

Next, dual modifier adaptation is applied with the lower bound on the inverse condition number of $\mathcal{U}(\mathbf{u})$, given in (2.41), as proposed for the dual ISOPE approach [12, 13]. The results are shown in Figure 4.22b. A lower bound of $\varphi = 0.4$ gives an adaptation almost identical to that of the gradient error bound in the first operating points. However, as soon as the neighborhood of the plant optimum is reached, the distance between the operating points decreases, and the gradient estimates become much less accurate. Furthermore, the feasible regions given by the condition number constraint decrease proportionally with the distance between points. This prevents the approach from taking large steps in the wrong direction, but it also makes it less suitable for tracking a changing optimum.

4.3 Summary

This chapter has developed a modifier-adaptation methodology in the context of real-time optimization. Unlike two-step approaches that rely

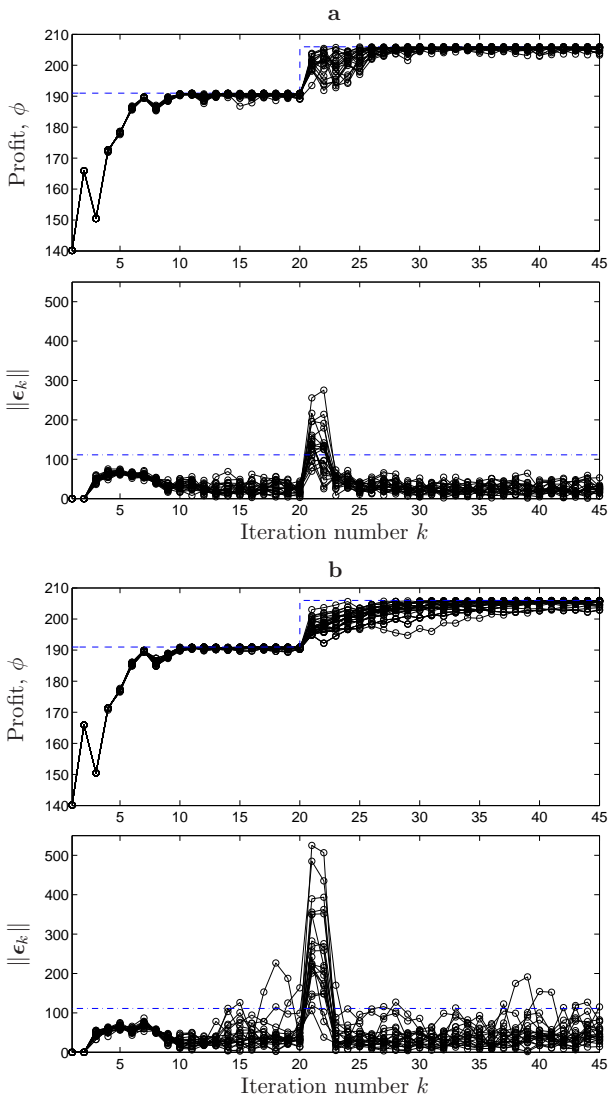


Fig. 4.22. (a) Dual modifier adaptation with upper bound on the gradient error norm. (b) Dual modifier adaptation with bound on the condition number. Dashed line: Optimal profit for the plant. Dash-dotted line: $\mathcal{E}^U = 111.2$.

on parameter estimation to adapt the parameters of a process model, modifier adaptation adjusts the optimization problem by adapting linear modifier terms in the cost and constraint functions. These modifiers are based on the differences between the measured and predicted values of (i) the constraints, and (ii) the constraint and cost gradients, i.e., quantities that are involved in the NCO. The adaptation of these modifiers in the model-based optimization is such that a point satisfying the *plant NCO* is reached upon convergence.

Various aspects of modifier adaptation have been discussed and illustrated through numerical examples, including local convergence conditions and model adequacy. The rather restrictive model-adequacy conditions of two-step approaches can be relaxed considerably, which represents a clear advantage of modifier adaptation. Variants of the modifier-adaptation scheme have also been presented, including an alternative filtering and an alternative modification of the process model. The differences and similarities with ISOPE and other approaches existing in the literature have been discussed.

The experimental application to a three-tank system in Subsection 4.1.8 shows the applicability of the approach. The major difficulty lies in the estimation of the experimental gradients. In this application, the constraint gradient was estimated using a finite-difference approach. However, other approaches available in the literature, such as those mentioned in Subsection 1.2.6, can be used as well. In this experimental application, improvement in optimality is observed upon correcting the constraint gradient predicted by the model. However, the perturbations used to estimate the gradients result in constraint violation within the RTO iterations. As discussed in Subsection 4.1.8, this feasibility loss could be avoided but at the expense of a loss in overall optimality with respect to constraint adaptation alone. In such a situation, it is advisable not to update the gradient modifiers at each iteration, but at a slower rate.

In Subsection 4.2.1 a rigorous upper bound on the gradient error norm was developed, which is used for positioning the new operating point with respect to past ones. This constraint takes into account the effect of truncation errors and measurement noise. For both error components, a rather conservative approach is adopted. The evaluation of the constraint requires the selection of two parameters corresponding to the function $\Psi(\mathbf{u})$ for which the gradient is being estimated: δ , which is representative of the level of measurement noise, and \mathbf{d}_2 , which is an upper bound on the curvature of $\Psi(\mathbf{u})$. The evaluation of the worst-case

scenario for the measurement noise component requires the evaluation of a number of distances between complement affine subspaces. The number of distances to evaluate quickly increases with the number of inputs. Also, the resulting upper bound on the gradient error norm becomes more conservative with increasing number of inputs, as the worst-case scenario for the measurement noise component will become less likely to happen.

Based on the upper bound on the gradient error, a dual modifier-adaptation algorithm was presented in Section 4.2. The Williams-Otto reactor case study has demonstrated the potential of the dual approach that uses several previous points for estimating the experimental gradient. In this application, dual modifier adaptation performed better than modifier adaptation using the FFD approach. Similar observations were made in [12, 13] when comparing dual ISOPE with standard ISOPE using different examples. However, it cannot be affirmed that, in every application, dual modifier adaptation will overperform the FFD approach. The performance will depend on the plant-model mismatch, the noise level and the value of d_2 , as well as on the gains used to filter the modifiers. Also, in the Williams-Otto reactor case study, the proposed dual modifier adaptation with a bound on the gradient error produced more accurate gradient estimates than dual modifier adaptation with a bound on the condition number, and thus it appears to be more suitable for tracking a changing optimum.

Future research could consider the combination of dual modifier adaptation with MPC constraint control, as was done in Section 3.3 for constraint adaptation alone. The combination of dual modifier adaptation with MPC is complicated by the fact that the optimal inputs evaluated at the RTO level are no longer applied directly to the plant, but are determined by the MPC controller, which might result in violation of the constraint on the gradient error norm.

Conclusions

5.1 Summary

In order for a model-based RTO approach to track the plant optimum, the process model should be able to predict accurately the constraint values as well as the constraint and cost gradients for the whole range of operating conditions and process disturbances. The classical two-step approach of RTO typically relies on a parameter estimation problem to update the parameters θ of a (typically nonlinear) first-principles model of the plant. Parameter estimation is complicated by plant-model mismatch and the lack of input excitation. Even if additional excitation may increase the number of identifiable parameters, the parameterization θ is itself not aimed at matching the KKT-related quantities \mathcal{C}_p , and the performance of the two-step approach heavily relies on the accuracy of the model. Hence, in the presence of plant-model mismatch and unmeasured disturbances, the modifier-adaptation approach is appealing since it does not require update of the process model via parameter estimation. In modifier-adaptation, the handles for adaptation are the modifiers Λ , which are updated based on the difference $\mathcal{C}_p - \mathcal{C}$ between the measured and predicted KKT-related quantities \mathcal{C}_p and \mathcal{C} . This way, there is a one-to-one correspondence between the modifiers Λ and the KKT-related quantities \mathcal{C}_p , thus resulting in a square decoupled adaptation problem designed to enforce KKT matching upon convergence. Modifier-adaptation permits to significantly relax model requirements. The price to pay is the need to estimate experimental gradients on-line.

In this thesis, the gradients are estimated by numerical approximation using $(n_u + 1)$ operating points. Hence, significant gradient estimates can be obtained by this approach provided the frequency of meaningful disturbances affecting the plant is sufficiently low with respect to the time required for the plant to reach a steady state $(n_u + 1)$ times. This clearly limits the applicability of this approach, in particular for systems with a large number of inputs. Modifier adaptation was applied to an experimental three-tank system in Subsection 4.1.8. The application was successful in that the correction of the constraint gradient permitted to attain an operating point with improved cost with respect to constraint adaptation alone. However, the finite-difference perturbations applied at each RTO iteration in order to estimate the gradient resulted in constraint violation and were detrimental to the overall optimality. Modifier adaptation might not scale to large-scale problems as easily as the two-step approach or constraint adaptation alone, since the difficulty in estimating experimental gradients from past operating points increases with the number of inputs. This re-emphasizes the need to investigate alternative ways of estimating experimental gradients.

For systems with sufficiently low variability, where the gradient estimates are meaningful, the dual modifier-adaptation approach studied in this thesis estimates the gradients using the past operating points generated by the RTO optimizer, rather than applying FFD perturbations at each RTO iteration. The proposed dual modifier adaptation pays attention to the accuracy with which the gradients are estimated. The results of the Williams-Otto reactor case study indicate that this approach, wherein the gradient error norm is upper bounded, produces more accurate gradient estimates than with simply bounding the condition number of $\mathcal{U}(\mathbf{u})$, i.e. a measure of the relative position of the successive inputs. In addition, the proposed scheme seems more capable of tracking a changing optimum. Although, in this work, the gradients are estimated at each RTO iteration, it is also possible to adapt the gradient modifiers less frequently than the constraint offsets, in particular for problems where the solution is largely determined by active constraints.

Without updating the gradients, constraint adaptation enforces feasibility but not optimality. Nevertheless, for a large number of optimization problems, for which the solution is mostly determined by the constraints, constraint adaptation can provide fast improvement, and handle changes in the active set. Furthermore, constraint adaptation

scales easily to large-scale problems, and can be combined in a convenient way with a constraint controller at the process control level, if the required online measurements are available. The applicability of constraint adaptation, with and without the constraint controller, was illustrated through the RTO of a SOFC system in Subsection 3.3.3.

Overall, this thesis has formalized the idea of using measurements to adapt the optimization problem in response to plant-model mismatch, following the paradigm of modifier adaptation, and has linked different work in the field of RTO.

5.2 Perspectives

Several theoretical and practical issues concerning the constraint-adaptation and modifier-adaptation schemes require further investigation, as well as possible extensions:

- It has been assumed throughout this thesis that the cost and constraint values of the plant can be directly evaluated from plant measurements at each RTO iteration. If some of these variables are not measured, a possible strategy is to use observers to estimate them based on the available measurements. For unmeasured constraints, it is possible to apply conservative constraint backoffs in order to prevent them from becoming violated.
- Necessary conditions of convergence have been formulated for the constraint-adaptation and modifier-adaptation algorithms. However, these conditions are inherently local and are not sufficient for convergence. Future research will consider sufficient conditions of convergence for both algorithms, applied to systems that exhibit sectorial nonlinearity.
- An approach for integrating constraint adaptation at the RTO level with MPC tailored to control the constraints at the process control level has been proposed in Subsection 3.3, and discussed from a methodological point of view. Future research is required to provide a deeper analysis of the integrated scheme. In order to implement RTO results, a two-stage MPC structure is frequently used in industry. The outer controller, which can be a linear program (LP-MPC) or a quadratic program (QP-MPC), uses a steady-state version of the prediction model used by the MPC to correct the setpoints passed from the RTO level to the MPC. The comparison of the advantages and disadvantages of the proposed integrated

scheme with LP-MPC and QP-MPC (see [88]) is an interesting subject of future research. While both approaches appear to provide offset-free control performance, the two-stage MPC requires an additional LP (or QP) level.

- Several aspects of the dual modifier-adaptation approach with a bound on the gradient error norm require further investigation. The application of the approach to constrained optimization problems was discussed in Subsection 4.2.3, where two strategies were proposed. There are still open questions regarding the selection of the parameters required to implement the approach, and on how to handle possible infeasibilities in the optimization problems. In addition, attention should be given to the way the gradient errors affect the loss in cost with respect to the plant infimum. As discussed in the conclusions of Chapter 4, the combination of dual modifier adaptation with constraint control needs also to be investigated.
- In this thesis, only steady-state perturbation methods have been considered for estimating the experimental gradients. Future research could consider the implementation of dynamic perturbation methods as well. In this context, the studies advanced in the ISOPE literature may serve as a point of departure [93].
- The constraint-adaptation approach has been extended to batch and semi-batch processes in [59]. Since batches are typically repeated, it is possible to devise run-to-run (also called batch-to-batch) optimization schemes by exploiting the knowledge from previous batches. The optimization of these processes typically involves solving a dynamic optimization problem for which the solution consists of time-varying input profiles. Dynamic optimization problems have two types of constraints: the path constraints limit the inputs and states during the batch, while the terminal constraints limit the outcome of the batch at final time. Hence, the path constraints are modified using time-varying modifiers, while the terminal constraints are modified using terminal modifiers. The analysis of the approach could be the subject of future work.

Finally, the implementation of constraint adaptation combined with constraint control to a SOFC system (see Subsection 3.3.3) has led to a collaboration project between the *Laboratoire d'Automatique (LA)* and the *Laboratoire d'Énergie Industrielle (LENI)* of EPFL. The project contemplates the experimental implementation of the approach to a SOFC system available at LENI's facilities.

A

Solid Oxide Fuel Cell Model

Energy Balance

The fuel and the oxidant (air) that enter the stack react electrochemically, releasing heat and electrical power. The energy balance for the stack is:

$$m_{\text{stack}} c_{p_{\text{stack}}} \frac{\partial T_{\text{stack}}}{\partial t} = -\Delta \dot{H}_{\text{gases}} - P - \dot{Q}_{\text{loss}} \quad (\text{A.1})$$

where the electrical power is given by:

$$P = U_{\text{cell}} N_{\text{cells}} I.$$

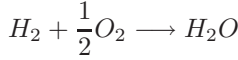
Only radiative heat loss from the stack is taken into account:

$$\dot{Q}_{\text{loss}} = A_{\text{stack}} \mathcal{F} \sigma_{\text{SB}} (T_{\text{stack}}^4 - T_{\text{furnace}}^4)$$

where \mathcal{F} is the transfer factor for the radiative heat exchange between the stack and the furnace calculated for the case of a body enclosed into another [53].

Mass Balance

For the mass balance, the species considered are H_2O and H_2 at the anode and O_2 and N_2 at the cathode. The only reaction taking place is the electrochemical oxidation of H_2 , for which the overall reaction is:



From Faraday's law, the amount of H_2 participating in the reaction is related to the current produced in the reaction as,

$$\dot{n}_{H_2, \text{reac}} = \frac{I N_{\text{cells}}}{2F}$$

The mass balances are formulated as,

Anode:

$$\dot{n}_{i, \text{an}, \text{out}} = \dot{n}_{i, \text{an}, \text{in}} + \nu_i \dot{n}_{H_2, \text{reac}}, \quad i = H_2, H_2O$$

Cathode:

$$\dot{n}_{j, \text{cath}, \text{out}} = \dot{n}_{j, \text{cath}, \text{in}} + \nu_j \dot{n}_{H_2, \text{reac}}, \quad j = O_2, N_2$$

where ν_i is the stoichiometric coefficient of component i in the reaction.

Electrochemical Model

The electrochemical model computes the cell potential and average current density as a function of the operating conditions, i.e., temperature, flow rates and gas compositions. The reversible cell voltage U_{Nernst} is computed from the change in Gibbs free enthalpy for the H_2 oxidation reaction as,

$$U_{\text{Nernst}} = \frac{-\Delta G_{\text{reaction}}}{n_e F}$$

This is the maximum amount of potential that can be delivered by the cell. The actual voltage is subject to overpotentials due to losses appearing during operation. The losses considered here are: ohmic losses due to the ionic resistance of the electrolyte and current collectors; activation losses due to charge transfer kinetics; diffusion losses due to concentration gradients between the electrode surface and the bulk flow; and losses due to the dissociation of the oxygen molecules into ions on the cathode surface. The effective cell potential U_{cell} is given by,

$$U_{\text{cell}} = U_{\text{Nernst}} - \eta_{\text{act}, \text{cath}} - \eta_{\text{ionic}, \text{elect}} - \eta_{\text{diss}, \text{cath}} - \eta_{\text{diff}, \text{an}} - \eta_{\text{diff}, \text{cath}}$$

The cathode activation overpotential is expressed by the Butler-Volmer equation [19]:

$$\eta_{\text{act,cath}} = \frac{R T_{\text{stack}}}{F} \sinh^{-1} \left(\frac{i}{2 i_{0,\text{cath}}} \right)$$

$$i_{0,\text{cath}} = \frac{2R T_{\text{stack}}}{F} k_{0,\text{cath}} \exp \left(\frac{-E_{\text{act,cath}}}{R T_{\text{stack}}} \right)$$

The anode overpotential is relatively small and is neglected.

The Ohmic overpotential is expressed as [69],

$$\eta_{\text{ionic,elect}} = i \left(\frac{h_{\text{elect}}}{\sigma_{\text{ionic,elect}}} \right)$$

$$\sigma_{\text{ionic,elect}} = \sigma_{0,\text{elect}} \exp \left(\frac{-E_{\text{elect}}}{R T_{\text{stack}}} \right)$$

The concentration overpotential in the anode is calculated as:

$$\eta_{\text{diff,an}} = -\frac{RT_{\text{stack}}}{2F} \ln(1 - (FU + FU_{\text{adj}}))$$

$$FU = \frac{\dot{n}_{\text{H}_2,\text{reac}}}{\dot{n}_{\text{H}_2,\text{an,in}}}$$

where FU is the Fuel Utilization factor, defined as the ratio of amount of H_2 consumed to the amount of H_2 at the inlet. FU_{adj} is an adjustment factor. The concentration overpotential in the cathode is calculated as:

$$\eta_{\text{diff,cath}} = -\frac{R T_{\text{stack}}}{2F} \ln \left(1 - \frac{FU}{\lambda_{\text{air}}} \right)$$

$$\lambda_{\text{air}} = \frac{2 \dot{n}_{\text{O}_2,\text{cath,in}}}{\dot{n}_{\text{H}_2,\text{an,in}}}$$

where λ_{air} is the excess air ratio, defined as the amount of oxygen to hydrogen in the feed over the stoichiometric ratio.

The overpotential loss due to the dissociation of oxygen at the cathode is expressed as:

$$\eta_{\text{diss,cath}} = R_{0,\text{cath}} \left(\frac{p_{\text{O}_2,\text{in}}}{p_0} \right)^{-0.5} \exp \left(\frac{E_{\text{diss,cath}}}{R T_{\text{stack}}} \right) i$$

The operating conditions are listed in Table A.1.

Table A.1. Fixed operating conditions

Fuel feed composition 3% H ₂ O, 97% H ₂	T_{in} (fuel and air)	750 °C
Air feed composition 21% O ₂ , 79% N ₂	T_{furnace}	780 °C

Table A.2. Parameter values

Kinetic parameters					
$E_{\text{act,cath}}$	1.5326×10^5	$\frac{\text{J}}{\text{mol}}$	$k_{0,\text{cath}}$	4.103×10^{11}	$\frac{1}{\Omega \text{ m}^2}$
E_{elect}	7.622×10^4	$\frac{\text{J}}{\text{mol}}$	$\sigma_{0,\text{elect}}$	1.63×10^4	$\frac{1}{\Omega \text{ m}}$
$R_{0,\text{cath}}$	9.2252×10^{-14}	$\Omega \text{ m}^2$	$E_{\text{diss,cath}}$	1.489×10^5	$\frac{\text{J}}{\text{mol}}$
Cell properties					
A_{active}	50×10^{-4}	m^2	A_{stack}	4.69×10^{-2}	m^2
m_{stack}	2.647	kg	$(m c_p)_{\text{stack}}$	2.33×10^2	$\frac{\text{J}}{\text{kg K}}$
N_{cells}	5		FU_{adj}	0.15	
\mathcal{F}	0.667				

Table A.3. SOFC Model Nomenclature

A_{active}	Active cell area (m^2)
A_{stack}	Area of stack exposed to the furnace (m^2)
$c_{\text{p,stack}}$	Heat capacity of stack ($\frac{\text{J}}{\text{kg K}}$)
E_{act}	Energy for reaction activation ($\frac{\text{J}}{\text{mol}}$)
E_{diss}	Activation energy for oxygen dissociation ($\frac{\text{J}}{\text{mol}}$)
E_{elect}	Activation energy for electrolyte conductivity ($\frac{\text{J}}{\text{mol}}$)
F	Faraday's constant
\mathcal{F}	Radiative heat exchange transfer factor
FU	Fuel utilization
FU_{adj}	FU adjustment factor
$\Delta G_{\text{reaction}}$	Free energy change for reaction ($\frac{\text{J}}{\text{mol}}$)
h	Thickness (m)
$\Delta \dot{H}_{\text{gases}}$	Enthalpy change for gases ($\frac{\text{J}}{\text{sec}}$)
I	Current (A)
i	Current density ($\frac{\text{A}}{\text{m}^2}$)
i_0	Exchange current density ($\frac{\text{A}}{\text{m}^2}$)
k_0	Pre-exponential factor for activation overpotential
k_B	Boltzmann constant
LHV	Lower heating value for H_2 ($\frac{\text{J}}{\text{mol}}$)
m_{stack}	Mass of materials of stack (kg)
n_e	Charge number of reaction
N_{cells}	Number of cells comprising the stack
P	Power produced by the stack (W)
P_{el}	Power density ($\frac{\text{W}}{\text{m}^2}$)
P_{blower}	Power consumed by blower (W)
p	Partial pressure ($\frac{\text{N}}{\text{m}^2}$)

p_0	Reference ambient pressure ($\frac{\text{N}}{\text{m}^2}$)
\dot{Q}_{air}	Volumetric flow rate of air ($\frac{\text{m}^3}{\text{s}}$)
\dot{Q}_{loss}	Heat loss from stack to furnace ($\frac{\text{J}}{\text{s}}$)
R	Universal gas constant
R_0	Pre-exponential factor for O_2 dissociation ($\Omega \text{ m}^2$)
T	Temperature (K)
U_{cell}	Cell potential (V)
U_{Nernst}	Nernst potential (V)

Greek letters

η_{act}	Activation overpotential (V)
η_{diff}	Diffusion overpotential (V)
η_{diss}	O_2 dissociation overpotential (V)
η_{ionic}	Ionic overpotential (V)
η	SOFC efficiency
λ_{air}	Excess air ratio
σ_{SB}	Stefan-Boltzmann constant
$\sigma_{0,\text{elect}}$	Ionic conductivity of electrolyte ($\frac{1}{\Omega \text{ m}}$)
ν	Stoichiometric coefficient

Subscripts

an	Anode
cath	Cathode
in	Inlet
out	Outlet
elect	Electrolyte

B

Affine Subspaces

In a n_u -dimensional space, a point is an affine subspace of dimension 0, a line is an affine subspace of dimension 1, and a plane is an affine subspace of dimension 2. An affine subspace of dimension $(n_u - 1)$ is called an hyperplane.

Hyperplane. An hyperplane in n_u -dimensional space is given by

$$n_1 u_1 + n_2 u_2 + \cdots + n_{n_u} u_{n_u} = b, \quad \text{or:} \quad \mathbf{n}^\top \mathbf{u} = b \quad (\text{B.1})$$

and divides the space into two half-spaces: $\mathbf{n}^\top \mathbf{u} \geq b$, and $\mathbf{n}^\top \mathbf{u} \leq b$.

Complement affine subspaces. Given a set of $(n_u + 1)$ points in a n_u -dimensional space, $\mathcal{S} := \{\mathbf{u}^1, \dots, \mathbf{u}^{n_u+1}\}$, a proper subset \mathcal{S}^A , i.e. $\mathcal{S}^A \subsetneq \mathcal{S}$, of $n_u^A \in \{1, \dots, n\}$ points generates an affine subspace of dimension $(n_u^A - 1)$:

$$\mathbf{u} = \mathbf{u}^1 + \lambda_{1,2} \frac{\mathbf{u}^1 - \mathbf{u}^2}{\|\mathbf{u}^1 - \mathbf{u}^2\|} + \cdots + \lambda_{1,n_u^A} \frac{\mathbf{u}^1 - \mathbf{u}^{n_u^A}}{\|\mathbf{u}^1 - \mathbf{u}^{n_u^A}\|} \quad (\text{B.2})$$

where the parameters $\lambda_{1,2}, \dots, \lambda_{1,n_u^A}$ represent distances from the point \mathbf{u}^1 in the directions $\mathbf{u}^1 - \mathbf{u}^2, \dots, \mathbf{u}^1 - \mathbf{u}^{n_u^A}$, respectively. The complement subset $\mathcal{S}^C := \mathcal{S} \setminus \mathcal{S}^A$ of $(n_u + 1 - n_u^A)$ points, generates the complement affine subspace of dimension $(n_u - n_u^A)$:

$$\begin{aligned} \mathbf{u} = & \mathbf{u}^{n_u^A+1} + \lambda_{n_u^A+1, n_u^A+2} \frac{\mathbf{u}^{n_u^A+1} - \mathbf{u}^{n_u^A+2}}{\|\mathbf{u}^{n_u^A+1} - \mathbf{u}^{n_u^A+2}\|} + \cdots \\ & \cdots + \lambda_{n_u^A+1, n+1} \frac{\mathbf{u}^{n_u^A+1} - \mathbf{u}^{n+1}}{\|\mathbf{u}^{n_u^A+1} - \mathbf{u}^{n+1}\|} \end{aligned} \quad (\text{B.3})$$

Distance between complement affine subspaces. In order to compute the distance between the complement affine subspaces (B.2) and (B.3), a vector \mathbf{n} that is normal to both subspaces is required:

$$\begin{aligned} & [\mathbf{u}^1 - \mathbf{u}^2 \ \dots \ \mathbf{u}^1 - \mathbf{u}^{n_u^A} \ \mathbf{u}^{n_u^A+1} - \mathbf{u}^{n_u^A+2} \ \dots] \\ & \quad \mathbf{u}^{n_u^A+1} - \mathbf{u}^{n_u+1}]^T \mathbf{n} = \mathbf{0}, \quad \text{or,} \quad \mathbf{U}\mathbf{n} = \mathbf{0}. \end{aligned} \quad (\text{B.4})$$

The matrix $\mathbf{U} \in \mathbb{R}^{(n_u-1) \times n_u}$ is of rank $(n_u - 1)$. Vector \mathbf{n} can be obtained by singular-value decomposition of \mathbf{U} .

Given a point \mathbf{u}^a that belongs to the affine subspace (B.2), a point \mathbf{u}^b that belongs to the complement affine subspace (B.3), and a vector \mathbf{n} that is normal to both complement affine subspaces, the distance l_{AC} between the complement affine subspaces is:

$$l_{AC} = \frac{|\mathbf{n}^T(\mathbf{u}^b - \mathbf{u}^a)|}{\|\mathbf{n}\|} \quad (\text{B.5})$$

References

- [1] P. Aguiar, C. S. Adjiman, and N. P. Brandon. Anode-supported intermediate-temperature direct internal reforming solid oxide fuel cell II. model-based dynamic performance and control. *J. Power Sources*, 147:136–147, 2005.
- [2] Y. Arkun and G. Stephanopoulos. Studies in the synthesis of control structures for chemical processes: Part IV. Design of steady-state optimizing control structures for chemical process units. *AIChE J.*, 26(6):975–991, 1980.
- [3] Y. Arkun and G. Stephanopoulos. Studies in the synthesis of control structures for chemical processes: Part V. Design of steady-state optimizing control structures for integrated chemical-plants. *AIChE J.*, 27(5):779–793, 1981.
- [4] T. Backx, O. Bosgra, and W. Marquardt. Integration of model predictive control and optimization of processes. In *ADCHEM 2000*, pages 249–260, Pisa, Italy, 2000.
- [5] W. Bamberger and R. Isermann. Adaptive on-line steady state optimization of slow dynamic processes. *Automatica*, 14:223–230, 1978.
- [6] B. Bank, J. Guddat, D. Klatte, B. Kummer, and K. Tammer. *Non-Linear Parametric Optimization*. Birkhäuser Verlag, Basel, 1983.
- [7] M. S. Bazaraa, H. D. Sherali, and C. M. Shetty. *Nonlinear Programming: Theory and Algorithms*. John Wiley and Sons, New Jersey, 3rd edition, 2006.
- [8] L. T. Biegler, I. E. Grossmann, and A. W. Westerberg. A note on approximation techniques used for process optimization. *Comp.*

- Chem. Eng.*, 9(2):201–206, 1985.
- [9] G. E. P. Box and N. R. Draper. *Evolutionary Operation. A Statistical Method for Process Improvement*. John Wiley, New York, 1969.
- [10] M. Brdyś, S. Chen, and P. D. Roberts. An extension to the modified two-step algorithm for steady-state system optimization and parameter estimation. *Int. J. Systems Sci.*, 17(8):1229–1243, 1986.
- [11] M. Brdyś and P. D. Roberts. Convergence and optimality of modified two-step algorithm for integrated system optimisation and parameter estimation. *Int. J. Systems Sci.*, 18(7):1305–1322, 1987.
- [12] M. Brdyś and P. Tatjewski. An algorithm for steady-state optimizing dual control of uncertain plants. In *Proc. 1st IFAC Workshop on New Trends in Design of Control Systems*, pages 249–254, Smolenice, Slovakia, 1994.
- [13] M. Brdyś and P. Tatjewski. *Iterative Algorithms for Multilayer Optimizing Control*. Imperial College Press, London UK, 2005.
- [14] R. C. M. Brekelmans, L. T. Driessen, H. L. M. Hamers, and D. den Hertog. Gradient estimation schemes for noisy functions. *J. Opt. Th. Appl.*, 126(3):529–551, 2005.
- [15] C. Brosilow and G. Q. Zhao. A linear programming approach to constrained multivariable process control. *Control Dyn. Syst.*, 27(3):141–181, 1988.
- [16] E. F. Camacho and C. Bordons. *Model Predictive Control*. Springer-Verlag, London, 1999.
- [17] B. Chachuat, A. Marchetti, and D. Bonvin. Process optimization via constraints adaptation. *J. Process Contr.*, 18:244–257, 2008.
- [18] B. Chachuat, B. Srinivasan, and D. Bonvin. Adaptation strategies for real-time optimization. *In revision: Comp. Chem. Eng.*, 2009.
- [19] S. H. Chan, K. A. Khor, and Z. T. Xia. A complete polarization model of a solid oxide fuel cell and its sensitivity to change of cell component thickness. *J. Power Sources*, 93:130–140, 2001.
- [20] C. Y. Chen and B. Joseph. On-line optimization using a two-phase approach: An application study. *Ind. Eng. Chem. Res.*, 26:1924–1930, 1987.
- [21] C. R. Cutler and R. T. Perry. Real time optimization with multivariable control is required to maximize profits. *Comp. Chem. Eng.*, 7(5):663–667, 1983.

- [22] C. R. Cutler and B. C. Ramaker. Dynamic matrix control - A computer control algorithm. In *Joint Automat. Contr. Conf.*, San Francisco, CA, 1980.
- [23] M. L. Darby and D. C. White. On-line optimization of complex process units. *Chem. Eng. Process.*, Oct:51–59, 1988.
- [24] A. Desbiens and A. A. Shook. IMC-optimization of a direct reduced iron phenomenological simulator. In *4th International Conference on Control and Automation*, pages 446–450, Montreal, Canada, 2003.
- [25] M. M. Diehl. *Real-Time Optimization for Large Scale Nonlinear Processes*. PhD thesis, Heidelberg University, 2001.
- [26] S. Diethelm, J. van Herle, Z. Wuillemin, A. Nakajo, N. Autissier, and M. Molinelli. Impact of materials and design on solid oxide fuel cell stack operation. *J. Fuel Cell Sci. Technol.*, 5(3):31003, 2008.
- [27] R. C. Durbeck. *Principles for Simplification of Optimizing Control Models*. PhD thesis, Case Western Reserve University, USA, 1965.
- [28] J. E. Ellis, C. Kambhampati, G. Sheng, and P. D. Roberts. Approaches to the optimizing control problem. *Int. J. Systems Sci.*, 19(10):1969–1985, 1988.
- [29] S. Engell. Feedback control for optimal process operation. *J. Process Contr.*, 17:203–219, 2007.
- [30] A. V. Fiacco. *Introduction to Sensitivity and Stability Analysis in Nonlinear Programming*, volume 165 of *Mathematics in Science and Engineering*. Academic Press, New York, 1983.
- [31] A. V. Fiacco and Y. Ishizuka. Sensitivity and stability analysis for nonlinear programming. *Ann. Oper. Res.*, 27:215–236, 1990.
- [32] W. R. Fisher, M. F. Doherty, and J. M. Douglas. The interface between design and control. 3. Selecting a set of controlled variables. *Ind. Eng. Chem. Res.*, 27:611–615, 1988.
- [33] J. F. Forbes and T. E. Marlin. Model accuracy for economic optimizing controllers: the bias update case. *Ind. Eng. Chem. Res.*, 33:1919–1929, 1994.
- [34] J. F. Forbes and T. E. Marlin. Design cost: A systematic approach to technology selection for model-based real-time optimization systems. *Comp. Chem. Eng.*, 20:717–734, 1996.
- [35] J. F. Forbes, T. E. Marlin, and J. F. MacGregor. Model adequacy requirements for optimizing plant operations. *Comp. Chem. Eng.*, 18(6):497–510, 1994.

- [36] 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(6):701–712, 2005.
- [37] W. Gao and S. Engell. Iterative set-point optimization of batch chromatography. *Comp. Chem. Eng.*, 29:1401–1409, 2005.
- [38] C. E. Garcia and M. Morari. Optimal operation of integrated processing systems. Part I: Open-loop on-line optimizing control. *AIChE J.*, 27(6):960–968, 1981.
- [39] C. E. Garcia and M. Morari. Internal model control. 1. A unifying review and some new results. *Ind. Eng. Chem. Process Des. Dev.*, 21:308–323, 1982.
- [40] C. E. Garcia and M. Morari. Optimal operation of integrated processing systems. Part II: Closed-loop on-line optimizing control. *AIChE J.*, 30(2):226–234, 1984.
- [41] C. E. Garcia and M. Morari. Internal model control. 2. Design procedure for multivariable systems. *Ind. Eng. Chem. Process Des. Dev.*, 24:472–484, 1985.
- [42] P. E. Gill, W. Murray, and M. H. Wright. *Practical Optimization*. Academic Press, London, 2003.
- [43] J. Golbert and D. R. Lewin. Model-based control of fuel cells: (1) Regulatory control. *J. Power Sources*, 135:135–151, 2004.
- [44] J. Golbert and D. R. Lewin. Model-based control of fuel cells (2): Optimal efficiency. *J. Power Sources*, 173:298–309, 2007.
- [45] W. W. Hogan. Point-to-set maps in mathematical programming. *SIAM Rev.*, 15:591–603, 1973.
- [46] S.-S. Jang, B. Joseph, and H. Mukai. On-line optimization of constrained multivariable chemical processes. *AIChE J.*, 33(1):26–35, 1987.
- [47] F. Jurrado. Predictive control of solid oxide fuel cells using fuzzy Hammerstein models. *J. Power Sources*, 158:245–253, 2006.
- [48] C. Kambhampati, J. E. Ellis, and P. D. Roberts. Generalization of Integrated System Optimization and Parameter Estimation Techniques. *Automatica*, 25(2):307–310, 1989.
- [49] 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. Eng.*, 28:2569–2581, 2004.

- [50] B. Kouvaritakis and M. Cannon. *Nonlinear Predictive Control Theory and Practice*. IEE Control Engineering Series 61. The Institution of Electrical Engineers, London, UK, 2001.
- [51] D. Larrain, J. Van herle, and D. Favrat. Simulation of SOFC stack and repeat elements including interconnect degradation and anode reoxidation risk. *J. Power Sources*, 161:392–403, 2006.
- [52] T. Larsson and S. Skogestad. Plantwide control - A review and a new design procedure. *Modeling, Identification and Control*, 21(4):209–240, 2000.
- [53] J. H. Lienhard IV and J. H. Lienhard V. *A Heat Transfer Text-book*. Phlogiston Press, 3rd edition, 2003.
- [54] C. Loeblein and J. D. Perkins. Economical analysis of different structures of on-line process optimization systems. *Comp. Chem. Eng.*, 20:S551–S556, 1996.
- [55] M. L. Luyben, B. D. Tyreus, and W. L. Luyben. Plantwide control design procedure. *AIChE J.*, 43(12):3161–3174, 1997.
- [56] A. Maarleveld and J. E. Rijnsdorp. Constraint control on distillation columns. *Automatica*, 6:51–58, 1970.
- [57] J. M. Maciejowski. *Predictive Control with Constraints*. Prentice Hall, New Jersey, 2002.
- [58] M. Mansour and J. E. Ellis. Comparison of methods for estimating real process derivatives in on-line optimization. *App. Math. Modelling*, 27:275–291, 2003.
- [59] A. Marchetti, B. Chachuat, and D. Bonvin. Batch process optimization via run-to-run constraints adaptation. In *ECC*, Kos, Greece, 2007.
- [60] A. Marchetti, B. Chachuat, and D. Bonvin. Real-time optimization of continuous processes via constraints adaptation. In *IFAC DYCOPS-8*, Cancun, Mexico, 2007.
- [61] A. Marchetti, B. Chachuat, and D. Bonvin. Real-time optimization via adaptation and control of the constraints. In *18th European Symposium on Computer Aided Process Engineering, ESCAPE 18*, Lyon, France, 2008.
- [62] A. Marchetti, A. Gopalakrishnan, L. Tsikonis, A. Nakajo, Z. Wullemmin, B. Chachuat, J. Van herle, and D. Bonvin. Real-time optimization of a solid oxide fuel cell stack. *J. Fuel Cell Sci. Technol. In preparation*, 2009.
- [63] T. E. Marlin and A. N. Hrymak. Real-time operations optimization of continuous processes. In *AIChE Symposium Series - CPC-V*, volume 93, pages 156–164, 1997.

- [64] M. Mercangöz and F. J. Doyle III. Real-time optimization of the pulp mill benchmark problem. *Comp. Chem. Eng.*, 32:789–804, 2008.
- [65] M. Morari, Y. Arkun, and G. Stephanopoulos. Studies in the synthesis of control structures for chemical processes. Part I: Formulation of the problem. Process decomposition and the classification of the control tasks. Analysis of the optimizing control structures. *AIChE J.*, 26(2):220–232, 1980.
- [66] F. Mueller, F. Jabbaria, R. Gaynor, and J. Brouwer. Novel solid oxide fuel cell system controller for rapid load following. *J. Power Sources*, 172:308–323, 2007.
- [67] Z. Nagy, R. Findeisen, M. Diehl, F. Allgöwer, H. G. Bock, S. Agachi, J. P. Schlöder, and D. Leineweber. Real-time feasibility of nonlinear predictive control for large scale processes - a case study. In *American Control Conference*, Chicago, Illinois, 2000.
- [68] A. Nakaïo, Z. Wullemin, J. Van herle, and D. Favrat. Simulation of thermal stresses in anode-supported solid oxide fuel cell stacks. Part I: Probability of failure of the cells. *J. Power Sources*, *In press*, 2009.
- [69] J.-H. Park and R. N. Blumenthal. Electronic transport in 8 mole percent. $Y_2O_3.ZrO_2$. *J. Electrochem. Soc.*, 136:2867–2876, 1989.
- [70] S. J. Qin and T. A. Badgwell. An overview of industrial model predictive technology. In *AIChE Symposium Series - CPC-V*, volume 93, pages 232–256, 1997.
- [71] J. Richalet, A. Rault, J. L. Testud, and J. Papon. Model predictive heuristic control: Application to industrial processes. *Automatica*, 14(2):413–428, 1978.
- [72] P. D. Roberts. An algorithm for steady-state system optimization and parameter estimation. *J. System Science*, 10:719–734, 1979.
- [73] P. D. Roberts. Coping with model-reality differences in industrial process optimisation - a review of integrated system optimisation and parameter estimation (ISOPE). *Computers in Industry*, 26:281–290, 1995.
- [74] P. D. Roberts and T. W. Williams. On an algorithm for combined system optimisation and parameter estimation. *Automatica*, 17(1):199–209, 1981.
- [75] R. T. Rockafellar. *Convex Analysis*. Princeton University Press, Princeton, New Jersey, 1997.

- [76] P. O. M. Scokaert and J. B. Rawlings. Feasibility issues in linear model predictive control. *AIChE J.*, 45(8):1649–1659, 1999.
- [77] S. E. Sequeira, M. Graells, and L. Puigjaner. Real-time evolution for on-line optimization of continuous processes. *Ind. Eng. Chem. Res.*, 41(7):1815–1825, 2002.
- [78] S. Skogestad. Self-optimizing control: The missing link between steady-state optimization and control. *Comp. Chem. Eng.*, 24:569–575, 2000.
- [79] B. Srinivasan, L. T. Biegler, and D. Bonvin. Tracking the necessary conditions of optimality with changing set of active constraints using a barrier-penalty function. *Comp. Chem. Eng.*, 32(3):572–579, 2008.
- [80] G. Stephanopoulos and C. Ng. Perspectives on the synthesis of plant-wide control structures. *J. Process Contr.*, 10:97–111, 2000.
- [81] W. Tadej and P. Tatjewski. Analysis of an ISOPE-type dual algorithm for optimizing control and nonlinear optimization. *Int. J. Appl. Math. Comput. Sci.*, 11(2):429–457, 2001.
- [82] P. Tatjewski. Iterative optimizing set-point control - The basic principle redesigned. In *15th IFAC World Congress*, Barcelona, Spain, 2002.
- [83] P. Tatjewski, M. A. Brdyś, and J. Duda. Optimizing control of uncertain plants with constrained feedback controlled outputs. *Int. J. Control*, 74(15):1510–1526, 2001.
- [84] T. Tosukhowong. *Dynamic Real-Time Optimization and Control of an Integrated Plant*. PhD thesis, Georgia Institute of Technology, 2006.
- [85] T. J. Williams and R. E. Otto. A generalized chemical processing model for the investigation of computer control. *AIEE Trans.*, 79:458, 1960.
- [86] X.-J. Wu, X.-J. Zhu, G.-Y. Cao, and H.-Y. Tu. Predictive control of SOFC based on a GA-RBF neural network model. *J. Power Sources*, 179:232–239, 2008.
- [87] Z. Wullemmin, N. Autissier, M.-T. Luong, J. Van herle, and D. Favrat. Modeling and study of the influence of sealing on a solid oxide fuel cell. *J. Fuel Cell Sci. Technol.*, 5:011016, 2008.
- [88] C.-M. Ying and B. Joseph. Performance and stability analysis of LP-MPC and QP-MPC cascade control systems. *AIChE J.*, 45(7):1521–1534, 1999.

- [89] W. S. Yip and T. E. Marlin. Multiple data sets for model updating in real-time operations optimization. *Comp. Chem. Eng.*, 26:1345–1362, 2002.
- [90] W. S. Yip and T. E. Marlin. Designing plant experiments for real-time optimization systems. *Control Engineering Practice*, 11:837–845, 2003.
- [91] W. S. Yip and T. E. Marlin. The effect of model fidelity on real-time optimization performance. *Comp. Chem. Eng.*, 28:267–280, 2004.
- [92] R. E. Young. Petroleum refining process control and real-time optimization. *IEEE Contr. Syst. Mag.*, 26(6):73–83, 2006.
- [93] H. Zhang and P. D. Roberts. On-line steady-state optimisation of nonlinear constrained processes with slow dynamics. *Tran. Inst. Meas. Contr.*, 12(5):251–261, 1990.
- [94] H. Zhang and P. D. Roberts. Integrated system optimization and parameter estimation using a general form of steady-state model. *Int. J. Systems Sci.*, 22(10):1679–1693, 1991.
- [95] X. W. Zhang, S. H. Chan, H. K. Hob, 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*, 33:2355–2366, 2008.
- [96] Y. Zhang. General robust-optimization formulation for nonlinear programming. *J. Opt. Th. Appl.*, 132(1):111–124, 2007.
- [97] Y. Zhang and J. F. Forbes. Extended design cost: A performance criterion for real-time optimization systems. *Comp. Chem. Eng.*, 24:1829–1841, 2000.
- [98] Y. Zhang and J. F. Forbes. Performance analysis of perturbation-based methods for real-time optimization. *The Canadian J. Chem. Eng.*, 84:209–218, 2006.
- [99] S. Zhijiang, W. Jinlin, and Q. Jixin. Real-time optimization of acetaldehyde production process. *Dev. Chem. Eng. Mineral Process*, 13(3-4):1–10, 2005.

Curriculum Vitae and List of Publications

Alejandro Gabriel Marchetti

Argentine citizen, single

Born on February 18, 1975, in Santa Fe, Argentina

Education

- 2003 - 2009: **PhD Thesis**, Laboratoire d'Automatique, École Polytechnique Fédérale de Lausanne, Switzerland. Director: Prof. D. Bonvin; Supervisor: Dr. B. Chachuat.
- 2002 - 2004: **Graduate Research Work** as scholarship holder of the Swiss Confederation. Laboratoire d'Automatique, École Polytechnique Fédérale de Lausanne, Switzerland. Director: Prof. D. Bonvin; Supervisor: Dr. B. Srinivasan.
- 1995 - 2002: **Diploma in Chemical Engineering**, Universidad Nacional del Litoral (UNL), Santa Fe, Argentina.
- 1988 - 1994: **Chemical Technician** and High School Education, Escuela Industrial Superior, UNL, Santa Fe, Argentina.

Publications

Journal Papers

Published

- A. Marchetti, B. Chachuat, and D. Bonvin. Modifier-Adaptation Methodology for Real-Time Optimization. *Ind. Eng. Chem. Res.*, 2009. In press.
- B. Chachuat, A. Marchetti, and D. Bonvin. Process Optimization via Constraints Adaptation. *J. Process Contr.*, 18(3-4):244–257, 2008.
- C. Welz, B. Srinivasan, A. Marchetti, D. Bonvin, and N. L. Ricker. Evaluation of Input Parameterization for Batch Process Optimization. *AIChE Journal.*, 52(9):3155–3163, 2006.

Pending

- A. Marchetti, A. Gopalakrishnan, L. Tsikonis, A. Nakajo, Z. Wuillemin, B. Chachuat, J. Van herle, and D. Bonvin. Real-Time Opti-

mization of a Solid Oxide Fuel Cell Stack. *J. Fuel Cell Sci. Technol.*, 2009. Submitted.

Conference Papers

- A. Marchetti, B. Chachuat, and D. Bonvin. Real-Time Optimization with Estimation of Experimental Gradient. In *IFAC Symp. ADCHEM*, Istanbul, Turkey, 2009.
- A. Marchetti, B. Chachuat, and D. Bonvin. Real-Time Optimization via Adaptation and Control of the Constraints. In *18th European Symposium on Computer Aided Process Engineering*, 393-398, Lyon, France, 2008.
- A. Marchetti, B. Chachuat, and D. Bonvin. Batch Process Optimization via Run-to-run Constraints Adaptation. In *European Control Conference ECC'07*, Kos, Greece, 2007.
- A. Marchetti, B. Chachuat, and D. Bonvin. Real-Time Optimization of Continuous Processes via Constraints Adaptation. In *8th International IFAC Symp. DYCOPS*, Vol. 1, 45-50, Cancún, Mexico, 2007.
- A. Marchetti, A. Amrhein, B. Chachuat, and D. Bonvin. Scale-up of Batch Processes via Decentralized Control. In *IFAC Symp. ADCHEM*, Gramado, Brazil, 2006.
- A. Marchetti, B. Srinivasan, D. Bonvin, S. Elgue, L. Prat, and M. Cabassud. Measurement-based Run-to-run Optimization of a Batch Reaction-distillation System. In *15th European Symposium on Computer Aided Process Engineering*, Barcelona, Spain, 2005.
- C. Welz, B. Srinivasan, A. Marchetti, D. Bonvin, and N. L. Ricker. Validation of a Solution Model for the Optimization of a Binary Batch Distillation Column. In *American Control Conference*, Portland, USA, 2005.
- A. Marchetti, B. Srinivasan, D. Bonvin, S. Elgue, L. Prat, and M. Cabassud. Validation du Modèle de la Solution pour l'Optimisation d'une Colonne de Distillation Réactive. In *10ème Congrès de la Société Française de Génie des Procédés*, Toulouse, France, 2005.