Fortschritt-Berichte VDI

Reihe 3

Verfahrenstechnik

Nr. 952

M.Sc. Xiao Zhao,
Xinjiang

Reactor network synthesis with guaranteed robust performance

Berichte aus der
Aachener Verfahrenstechnik - Prozesstechnik

RWTH Aachen University

https://doi.org/10.51202/9783186952035 - Generiert durch IP 216.73.216.36, am 18.01.2026, 21:11:22. © Urheberrechtlich geschützter Inhalt. Ohne gesonderte
Erlaubnis ist jede urheberrechtliche Nutzung untersagt, insbesondere die Nutzung des Inhalts im Zusammenhang mit, für oder in KI-Systemen, KI-Modellen oder Generativen Sprachmodellen.

https://doi.org/10.51202/9783186952035 - Generiert durch IP 216.73.216.36, am 18.01.2026, 21:11:22. © Urheberrechtlich geschützter Inhalt. Ohne gesonderte
Erlaubnis ist jede urheberrechtliche Nutzung untersagt, insbesondere die Nutzung des Inhalts im Zusammenhang mit, für oder in KI-Systemen, KI-Modellen oder Generativen Sprachmodellen.

Reactor Network Synthesis With Guaranteed Robust Performance

Synthese von Reaktornetzwerken mit robust garantierten Eigenschaften

Von der Fakultät für Maschinenwesen der Rheinisch-Westfälischen Technischen Hochschule Aachen vorgelegte Dissertation zur Erlangung des akademischen Grades eines Doktors der Ingenieurwissenschaften

vorgelegt von

Xiao Zhao
aus
Xinjiang, China

Berichter: Universitätsprofessor Dr.-Ing. Wolfgang Marquardt
Universitätsprofessor Dr.-Ing. Martin Mönnigmann

Tag der mündlichen Prüfung: 18.05.2017

D82 (Diss. RWTH Aachen University)

https://doi.org/10.51202/9783186952035 - Generiert durch IP 216.73.216.36, am 18.01.2026, 21:11:22. © Urheberrechtlich geschützter Inhalt. Ohne gesonderte Erlaubnis ist jede urheberrechtliche Nutzung untersagt, insbesondere die Nutzung des Inhalts im Zusammenhang mit, für oder in KI-Systemen, KI-Modellen oder Generativen Sprachmodellen.

https://doi.org/10.51202/9783186952035 - Generiert durch IP 216.73.216.36, am 18.01.2026, 21:11:22. © Urheberrechtlich geschützter Inhalt. Ohne gesonderte Erlaubnis ist jede urheberrechtliche Nutzung untersagt, insbesondere die Nutzung des Inhalts im Zusammenhang mit, für oder in KI-Systemen, KI-Modellen oder Generativen Sprachmodellen.

Fortschritt-Berichte VDI

Reihe 3

Verfahrenstechnik

M.Sc. Xiao Zhao,
Xinjiang

Nr. 952

Reactor network synthesis with guaranteed robust performance

Berichte aus der
Aachener Verfahrenstechnik - Prozesstechnik

RWTH Aachen University

https://doi.org/10.51202/9783186952035 - Generiert durch IP 216.73.216.36, am 18.01.2026, 21:11:22. © Urheberrechtlich geschützter Inhalt. Ohne gesonderte Erlaubnis ist jede urheberrechtliche Nutzung untersagt, insbesondere die Nutzung des Inhalts im Zusammenhang mit, für oder in KI-Systemen, KI-Modellen oder Generativen Sprachmodellen.

Zhao, Xiao
Reactor network synthesis with guaranteed robust performance
Fortschr.-Ber. VDI Reihe 3 Nr. 952. Düsseldorf: VDI Verlag 2017.
184 Seiten, 24 Bilder, 19 Tabellen.
ISBN 978-3-18-395203-8, ISSN 0178-9503,
€ 67,00/VDI-Mitgliederpreis € 60,30.

Für die Dokumentation: Reactor network synthesis, Integration of process and control system design, Superstructure approach, Eigenvalue constraints, Mixed-integer nonlinear programming, Normal vector approach, Robust optimization

In this work a systematic model-based approach for reactor network synthesis problem with guaranteed robust dynamic performance will be presented. The work is based on the superstructure approach and aims to find an optimal process flowsheet with determined connection patterns of reactors, reactor types, design parameters and operating conditions. In comparison to the classical design methods, certain specified dynamic properties are guaranteed simultaneously under parametric uncertainty. Structural alternatives in the flowsheet, i.e., how reactors are interconnected, as well as in the control system, i.e., how controlled and manipulated variables are paired, are subject to design degrees of freedom. It is allowed that idle reactors and controllers can appear in the reactor network superstructure, so that a fixed number of non-idle reactors and controllers does not have to be assumed as a priori. The optimal reactor network design in either open- or closed-loop is determined by solving a single optimization problem.

Bibliographische Information der Deutschen Bibliothek
Die Deutsche Bibliothek verzeichnet diese Publikation in der Deutschen Nationalbibliographie; detaillierte bibliographische Daten sind im Internet unter http://dnb.ddb.de abrufbar.

Bibliographic information published by the Deutsche Bibliothek
(German National Library)
The Deutsche Bibliothek lists this publication in the Deutsche Nationalbibliographie (German National Bibliography); detailed bibliographic data is available via Internet at http://dnb.ddb.de.

D82 (Diss. RWTH Aachen University, 2017)

© VDI Verlag GmbH · Düsseldorf 2017

Alle Rechte, auch das des auszugsweisen Nachdruckes, der auszugsweisen oder vollständigen Wiedergabe (Fotokopie, Mikrokopie), der Speicherung in Datenverarbeitungsanlagen, im Internet und das der Übersetzung, vorbehalten.

Als Manuskript gedruckt. Printed in Germany.

ISSN 0178-9503

ISBN 978-3-18-395203-8

https://doi.org/10.51202/9783186952035 - Generiert durch IP 216.73.216.36, am 18.01.2026, 21:11:22. © Urheberrechtlich geschützter Inhalt. Ohne gesonderte Erlaubnis ist jede urheberrechtliche Nutzung untersagt, insbesondere die Nutzung des Inhalts im Zusammenhang mit, für oder in KI-Systemen, KI-Modellen oder Generativen Sprachmodellen.

Acknowledgements

The present thesis originates from my study and work time at Aachener Verfahrenstechnik, Process Systems Engineering at RWTH Aachen University as scientific staff.

I would like to express my deep and sincere gratitude to my thesis advisor Prof. Dr.-Ing. Wolfgang Marquardt. His thoughtful guidance, his valuable suggestions, his encouragements and his open mind to new ideas are the essential prerequisites of finishing this thesis. I also wish to express my warm and sincere thanks to Prof. Dr.-Ing. Martin Mönnigmann for reviewing my thesis and Prof. Dr.-Ing. Uwe Reisgen to be the chairman of exam commission.

I would like to thank all the colleagues at the institute for the friendly and cooperative atmosphere. I am particularly thankful to Diego A. Muñoz for his great help during my starting phase; Maxim Stuckert, Ralf Hannemann, Yi Heng und Ionut Muntean for the fruitful discussions on mathematics, Sebastian Recker and Mirko Skiborowski for their important support in process modeling. At the end, I would like to thank Moritz Schmitz, Michael Wiedau, Nimet Kerimoglu, Ganzhou Wang, Matthias Johannink und Klaus Stockmann for the good time working together in the same office.

Finally, I owe my loving thanks to my wife Rui and my daughter Meitang for their love and support. I would also like to thank my parents for their love and continuous support.

Aachen, in May 2017

https://doi.org/10.51202/9783186952035 - Generiert durch IP 216.73.216.36, am 18.01.2026, 21:11:22. © Urheberrechtlich geschützter Inhalt. Ohne gesonderte
Erlaubnis ist jede urheberrechtliche Nutzung untersagt, insbesondere die Nutzung des Inhalts im Zusammenhang mit, für oder in KI-Systemen, KI-Modellen oder Generativen Sprachmodellen.

https://doi.org/10.51202/9783186952035 - Generiert durch IP 216.73.216.36, am 18.01.2026, 21:11:22. © Urheberrechtlich geschützter Inhalt. Ohne gesonderte
Erlaubnis ist jede urheberrechtliche Nutzung untersagt, insbesondere die Nutzung des Inhalts im Zusammenhang mit, für oder in KI-Systemen, KI-Modellen oder Generativen Sprachmodellen.

Contents

Notation VIII

1 Introduction 1
1.1 Reactor network synthesis . . . 1
1.1.1 Motivation . . . 1
1.1.2 Task . . . 1
1.1.3 Methods . . . 2
1.2 Design of decentralized control systems . . . 3
1.2.1 Tasks . . . 4
1.2.2 Methods . . . 5
1.3 Simultaneous process and control system design . . . 7
1.3.1 Motivation . . . 7
1.3.2 Methods . . . 8
1.4 Content and goals of this work . . . 10
1.4.1 Overview . . . 10
1.4.2 New features of this work . . . 14

2 Some Preliminaries 17
2.1 Dynamic systems . . . 17
2.2 Eigenvalue and spectral abscissa functions . . . 19
2.3 Lyapunov stability . . . 23
2.4 Dynamic response . . . 25
2.5 Extension to differential-algebraic systems . . . 28

3 Open-loop reactor network synthesis 30
3.1 Structured modeling of reactor networks . . . 30
3.1.1 A structured representation of reactor network models . . . 31
3.1.2 Models of Subsystem 1 to N . . . 34
3.1.3 Models of subsystems $N + 1$ and $N + 2$ . . . 36
3.1.4 Modeling flow connections . . . 37
3.1.5 A dynamic model of the network . . . 38
3.1.6 Idle reactors in open-loop reactor networks . . . 39
3.2 Problem formulation . . . 40
3.2.1 Eigenvalue constraint for open-loop reactor network synthesis . . . 40
3.2.2 Continuity analysis of the proposed eigenvalue constraint . . . 41
3.2.3 A direct problem formulation . . . 43
3.2.4 Problem reformulation . . . 44
3.3 Summary . . . 46

4 Simultaneous design of reactor network and its decentralized control system 47

https://doi.org/10.51202/9783186952035 - Generiert durch IP 216.73.216.36, am 18.01.2026, 21:11:22. © Urheberrechtlich geschützter Inhalt. Ohne gesonderte
Erlaubnis ist jede urheberrechtliche Nutzung untersagt, insbesondere die Nutzung des Inhalts im Zusammenhang mit, für oder in KI-Systemen, KI-Modellen oder Generativen Sprachmodellen.

4.1 Modeling of reactor networks with decentralized control structure 47
4.1.1 A closed-loop reactor network model 47
4.1.2 Complementarity constraints for control structure selection 50
4.1.3 Idle reactors and controllers 51
4.1.4 Structural constraints 53
4.1.5 Structural properties of the closed-loop model 54
4.2 Problem formulation 55
4.2.1 Eigenvalue constraint for simultaneous reactor network and control system synthesis 56
4.2.2 Problem formulation 57
4.3 Summary 58

5 Solution methods 60
5.1 Discrete-continuous optimization 60
5.1.1 Mixed-integer nonlinear program 60
5.1.2 General concepts to solve MINLP 60
5.1.3 Solution methods for convex MINLP 61
5.1.4 Solution methods for non-convex MINLP 62
5.1.5 Generalized disjunctive programming 63
5.2 Mathematical programs with complementarity constraints 64
5.2.1 MPCC versus NLP 65
5.2.2 Solution methods for MPCC 67
5.2.3 Relationship to MINLP 71
5.3 Semi-infinite programming 72
5.3.1 Local solution methods for SIP 73
5.3.2 Global solution methods for SIP 79
5.3.3 A robust design method: Normal vector approach 82
5.4 Eigenvalue optimization 89
5.4.1 Relation to semi-definite programming 90
5.4.2 Solution methods for EVO: Non-smooth optimization 91
5.4.3 Solution methods for EVO: Smoothing techniques 94
5.5 Challenges of solving the derived reactor network synthesis problem 99
5.6 A proposed two-step hybrid solution method 100
5.6.1 Step 1: Mixed-integer problem without uncertainty 101
5.6.2 Step 2: Robust optimization problem 102
5.7 Implementation 104

6 Case study of allyl chloride production 107
6.1 Open-loop reactor network design with robust stability 107
6.1.1 Problem setting 107
6.1.2 Design results 110
6.2 Simultaneous reactor network and control system design for fast response . 120
6.2.1 Closed-loop reactor network modeling 120
6.2.2 Problem setting 122
6.2.3 Results and discussion 123
6.2.4 Comparison with established sequential design 127
6.3 Computational experience 130

https://doi.org/10.51202/9783186952035 - Generiert durch IP 216.73.216.36, am 18.01.2026, 21:11:22. © Urheberrechtlich geschützter Inhalt. Ohne gesonderte Erlaubnis ist jede urheberrechtliche Nutzung untersagt, insbesondere die Nutzung des Inhalts im Zusammenhang mit, für oder in KI-Systemen, KI-Modellen oder Generativen Sprachmodellen.

7 Summary and outlook **133**
7.1 Contributions and summary of this work 133
7.2 Future research directions 135
7.2.1 Extensions regarding to the guaranteed robust dynamic properties . 135
7.2.2 Extensions regarding optimization methods 136
7.2.3 Extension to control structure selection for linear system 137
7.2.4 Other possible extensions and improvements 139

Appendices **141**

A Extension of the reactor network model to PFR **142**

B Proof of Proposition 3.2.1 **145**

C Proof of Proposition 3.2.2 **148**

D Parametric optimization problems **149**

E Proof of Theorem 5.3.10 **151**

Bibliography **158**

https://doi.org/10.51202/9783186952035 - Generiert durch IP 216.73.216.36, am 18.01.2026, 21:11:22. © Urheberrechtlich geschützter Inhalt. Ohne gesonderte Erlaubnis ist jede urheberrechtliche Nutzung untersagt, insbesondere die Nutzung des Inhalts im Zusammenhang mit, für oder in KI-Systemen, KI-Modellen oder Generativen Sprachmodellen.

Notation

We summarize the symbols for reactor network modeling and the derived problem formulations in Chapter 3 and 4. The symbols for the introduction part of nonlinear systems in Chapter 2 and the symbols for reviewing different numerical optimization methods shown in Chapter 5 are not presented here. The meaning of these symbols is always introduced together with the corresponding texts.

Symbols for the reaction example

A	propylene
B	allyl chloride
C	chlorine
c_A, c_B, c_C	concentration of component A, B or C
R	gas constant
a_1, a_2, a_3	reaction constants
H_1, H_2, H_3	heat of reaction per mol
c_p	heat capacity
T	temperature
V	reactor volume
L	reactor length
r_1, r_2, r_3	reaction rates
R_A, R_B, R_C	reaction rates for component A, B, C
$\dot{n}_A^0, \dot{n}_B^0, \dot{n}_C^0$	mole flowrates of inlets
$\dot{Q}^0$	energy flowrate of inlets
Q_h	energy duty of heat exchanger
$c_A^{sys}, c_B^{sys}, c_C^{sys}$	concentration in system's inlet
E^{sys}	energy density in system's inlet
N_d	number of discretized points of each PFR

Symbols for open-loop reactor network design

i	index of subsystems
j	index of outlet ports
k	index of inlet ports
N	total number of reactors
(i, j)	index of the j-th outlet port of subsystem i
(i, k)	index of the k-th inlet port of subsystem i
$l(i, j)$	index of an inlet port, which is connected to (i, j)
$h(i, k)$	index of an outlet port, which is connected to (i, k)

https://doi.org/10.51202/9783186952035 - Generiert durch IP 216.73.216.36, am 18.01.2026, 21:11:22. © Urheberrechtlich geschützter Inhalt. Ohne gesonderte Erlaubnis ist jede urheberrechtliche Nutzung untersagt, insbesondere die Nutzung des Inhalts im Zusammenhang mit, für oder in KI-Systemen, KI-Modellen oder Generativen Sprachmodellen.

$(i,j) \triangleright (i',k')$	pipe connection from (i,j) to (i',k')
$\bar{l}(i,j)$	index of a subsystem, one of its inlets is connected to (i,j)
$\bar{h}(i,k)$	index of a subsystem, one of its outlets is connected to (i,k)
N_c	number of necessary chemical components to model reactions
x_i	states (concentrations, temperature) of reactor i
$u_{i,k}$	component flowrates and energy flowrate through inlet port (i,k)
$q_{i,j}$	volumetric flowrate through the (i,j)-th outlet port
p_i	design parameters of reactor i
$f_i(\cdot)$	function for mass and energy balances of reactor i
$y_{i,j}$	component flowrates and energy flowrate through outlet port (i,j)
$g_{i,j}(\cdot)$	function for reactor's outlets
y_{sys}	component flowrates and energy flowrate in system's outlet
p_{sys}	molar concentration and energy density in the system's feed
$\mathcal{I}$	index set of all reactors
$\mathcal{I}_{id}$	index set of idle reactors
$\mathcal{I}_{nid}$	index set of non-idle reactors
J_{tot}	Jacobian matrix of the open-loop reactor network
J_{id}	Jacobian matrix of idle reactors
J_{nid}	Jacobian matrix of non-idle reactors
$\bar{J}$	a constructed matrix
c	predefined constant for the upper bound of eigenvalue constraints
$\alpha(\cdot)$	spectral abscissa function
D_o	definition domain of function $\alpha_{J_{nid}}(\cdot)$
P^*	steady state of the 2-reactor network example
φ	objective function in optimization
π_τ	vector of uncertain variables
$\bar{\pi}_\tau$	nominal values of uncertain variables
$\Delta\bar{\pi}_\tau$	uncertain range of uncertain variables
z_i	integer for the existence of reactor i
z	a vector of z_i, $i = 1, ..., N$
M	sufficiently large positive constant in big-M method
I	identity matrix
ψ_o	degrees of freedom of the open-loop model
ϵ	a small positive number

Symbols for simultaneous reactor network design and control

u	candidate MV of reactor network
y	candidate CV of reactor network
e	state variables of PI controllers
$\bar{u}$	offset values of u
$\bar{y}$	reference signals of y
$\bar{q}$	offset values of q
u_i	candidate MV of reactor i (elements of p_i)
d_i	equipment design parameters of reactor i (elements of p_i)

https://doi.org/10.51202/9783186952035 - Generiert durch IP 216.73.216.36, am 18.01.2026, 21:11:22. © Urheberrechtlich geschützter Inhalt. Ohne gesonderte Erlaubnis ist jede urheberrechtliche Nutzung untersagt, insbesondere die Nutzung des Inhalts im Zusammenhang mit, für oder in KI-Systemen, KI-Modellen oder Generativen Sprachmodellen.

u_{p1}	candidate MV, which belong to idle reactors
u_{p2}	candidate MV, which do not belong to idle reactors
n_c	dimension of y, i.e. total number of candidate CV
n_m	dimension of u, i.e. total number of candidate MV
(i, r)	index of the r-th candidate measurement of reactor i
	index of the (i, r)-th candidate PI controller
n_c^i	dimension of all candidate CV of reactor i
$e_{i,r}$	state variable of (i, r)-th PI controller
π	variables in ψ_o, which are not in u
v	location index for candidate MV
	location index for rows of control gain matrix K
w	location index for candidate CV
	location index for columns of control gain matrix K
$[u]_v$	v-th element in vector u
$[y]_m$	m-th element in vector y
Θ_i	index set for candidate MV of reactor i
$y_{i,r}$	r-th candidate CV of reactor i
$\phi_{i,r}(\cdot)$	function for candidate CV
$\varrho(\cdot, \cdot)$	function for transforming the subscripts of $[y]_m$ and $y_{(i,r)}$
K	proportional control gain matrix
$[K]_{v,w}$	(v, w)-th element in matrix K
K_v	vector of variables in matrix K
K^+, K^-, $\hat{K}$	auxiliary matrices for control structure selection
T	integral control gain matrix
$t_{i,r}$	integral control gain for state $e_{i,r}$
T_v	vector of variables in matrix T
ψ_c	degrees of freedom of the closed-loop model
$\mathcal{U}$	index set of all candidate MV
$\mathcal{U}_{id}$	index set of candidate MV, which are not subject to control
$\mathcal{U}_{nid}$	index set of candidate MV, which are subject to control
$\mathcal{C}$	index set of all PI controllers
$\mathcal{C}_{id}$	index set of idle PI controllers
$\mathcal{C}_{nid}$	index set of non-idle PI controllers
$z_{i,r}$	integer for the existence of the (i, r)-th PI controller
z_r	a vector of z_i (existence of reactors)
z_c	a vector of $z_{i,r}$ (existence of controllers)
x_{id}	states of idle reactors
x_{nid}	states of non-idle reactors
e_{id}	states of idle controllers
e_{nid}	states of non-idle controllers
J_{tot}	Jacobian matrix of the closed-loop reactor network model
J_{id}, J'_{id}, J_{nid}, J'_{nid}	Submatrices in J_{tot}
$F_{id}(\cdot)$	state functions of all idle reactors and controllers
$F_{nid}(\cdot)$	state functions of all non-idle reactors and controllers
$\bar{J}$	a constructed matrix

https://doi.org/10.51202/9783186952035 - Generiert durch IP 216.73.216.36, am 18.01.2026, 21:11:22. © Urheberrechtlich geschützter Inhalt. Ohne gesonderte Erlaubnis ist jede urheberrechtliche Nutzung untersagt, insbesondere die Nutzung des Inhalts im Zusammenhang mit, für oder in KI-Systemen, KI-Modellen oder Generativen Sprachmodellen.

Mathematical notations

$\mathbb{R}$	real line
$\mathbb{R}^n$	real n-dimensional space
$\mathbb{R}^n_+$	non-negative orthant of $\mathbb{R}^n$
$\mathbb{C}$	complex plane
$\mathcal{C}^k$	space of k-th order continuously differentiable functions
$\frac{\partial f}{\partial x}$	partial derivatives of function $f(x)$ to x
$\overline{B}$	topological closure of a set B

Acronyms

NLP	nonlinear optimization
MINLP	mixed-integer nonlinear optimization
MILP	mixed-integer linear optimization
MIDO	mixed-integer dynamic optimization
GDP	generalized disjunctive programming
MPCC	mathematical programs with complementarity constraints
MPEC	mathematical programs with equilibrium constraints
SIP	semi-infinite programming
GSIP	generalized semi-infinite programming
EVO	eigenvalue optimization
SDP	semi-definite programming
NSO	non-smooth optimization
DOF	degrees of freedom
SA	spectral abscissa
NVA	normal vector approach
CV	controlled variable
MV	manipulated variable
B&B	branch and bound (with respect to binary variables)
sB&B	spatial branch and bound
GBD	generalized bender's decomposition
VI	variational inequalities
MFCQ	Mangasarian Fromovitz constraint qualification
LICQ	linear independence constraint qualification
KKT	Karush Kuhn Tucker
SQP	sequential quadratic programming
NCP	nonlinear complementary problem
FB	Fischer-Burmeister
BL	bi-level
EPF	elementary process functions
AR	attainable region
PI	proportional-integral
RGA	relative gain array
SV	singular values

https://doi.org/10.51202/9783186952035 - Generiert durch IP 216.73.216.36, am 18.01.2026, 21:11:22. © Urheberrechtlich geschützter Inhalt. Ohne gesonderte
Erlaubnis ist jede urheberrechtliche Nutzung untersagt, insbesondere die Nutzung des Inhalts im Zusammenhang mit, für oder in KI-Systemen, KI-Modellen oder Generativen Sprachmodellen.

NI	Niederlinski index
SSV	structured singular value
MIMO	multi-input multi-output
ODE	ordinary differential equations
DAE	differential algebraic equations

https://doi.org/10.51202/9783186952035 - Generiert durch IP 216.73.216.36, am 18.01.2026, 21:11:22. © Urheberrechtlich geschützter Inhalt. Ohne gesonderte
Erlaubnis ist jede urheberrechtliche Nutzung untersagt, insbesondere die Nutzung des Inhalts im Zusammenhang mit, für oder in KI-Systemen, KI-Modellen oder Generativen Sprachmodellen.

Abstract

Typical continuous process flowsheets include reaction section, separation section and recycles. The reaction section is often the most important part of a chemical process, which may contain several interconnected reactors. The superstructure approach is a widely used model-based process design method for reactor network synthesis. It starts from a reactor network superstructure and uses mathematical models and optimization tools to select the best process design. The superstructure approach results in an optimal process flowsheet with determined connection patterns of reactors, reactor types, design parameters and operating conditions of each reactor.

In this work, a systematic model-based approach for reactor network synthesis problems with guaranteed robust dynamic performance will be presented. The work is based on the superstructure approach, but in comparison to the classical methods, not only economic optimality with respect to a static objective function, but also certain specified dynamic properties, i.e. dynamic stability and response speed, are guaranteed simultaneously under parametric uncertainty. Structural alternatives in the flowsheet, i.e., how reactors are interconnected, as well as in the control system, i.e., how controlled and manipulated variables are paired, are subject to design degrees of freedom. Moreover, it is allowed that idle reactors and controllers can appear in the reactor network superstructure, so that a fixed number of non-idle reactors and controllers does not have to be assumed as a priori. The optimal reactor network design in either open- or closed-loop is determined by solving a single optimization problem.

The proposed approach allows an integrated treatment of parametric uncertainties, which may either result from model uncertainties, such as reaction kinetic constants or heat transfer coefficients, or from process uncertainties, including slow disturbances in load or the quality of raw materials. A robust eigenvalue constraint to guarantee the robust performance of the designed reactor network is formulated. Efficient formulations of interconnecting reactors and novel complementarity-based constraints for control structure selection are proposed. The method results in a semi-infinite mixed-integer nonlinear optimization problem with complementarity constraints, disjunctions and a robust eigenvalue constraint. A hybrid two-step solution method is proposed to solve the synthesis problem, which integrates candidate solution algorithms of related optimization problems. The proposed solution method is applied to a case study of allyl chloride production with up to ten plug flow and continuous stirred tank reactors.

https://doi.org/10.51202/9783186952035 - Generiert durch IP 216.73.216.36, am 18.01.2026, 21:11:22. © Urheberrechtlich geschützter Inhalt. Ohne gesonderte Erlaubnis ist jede urheberrechtliche Nutzung untersagt, insbesondere die Nutzung des Inhalts im Zusammenhang mit, für oder in KI-Systemen, KI-Modellen oder Generativen Sprachmodellen.

Kurzfassung

Übliche kontinuierliche Prozesse enthalten einen Reaktionsteil, eine Trennsequenz und Rückführungen. Der Reaktionsteil stellt meist den wichtigsten Teil eines chemischen Prozesses dar, der aus vielen untereinander verknüpften Reaktoren bestehen kann. Der Überstrukturansatz beschreibt eine oft genutzte, modellgestützte Methode zur Erstellung von Reaktornetzwerken mit strukturellen Freiheitsgraden. Ausgehend von einer Überstruktur des Reaktornetzwerkes werden mathematische Modelle und Optimierungswerkzeuge genutzt, um den besten Prozessentwurf zu finden. Der Überstrukturansatz resultiert in einem optimalen Prozessfließbild mit festgelegten Verknüpfungen der Reaktoren eines bestimmten Reaktortyps sowie mit den zugehörigen Designparametern und Betriebsbedingungen für jeden Reaktor.

In dieser Arbeit wird ein systematischer, modellgestützter Ansatz für den Entwurf von Reaktornetzwerken mit garantiert robusten dynamischen Eigenschaften präsentiert. Die Arbeit basiert auf dem Überstrukturansatz. Im Vergleich zu konventionellen Methoden wird jedoch nicht nur die ökonomische Optimalität in Bezug auf eine statische Zielfunktion, sondern auch bestimmte spezifische dynamische Eigenschaften, insbesondere die dynamische Stabilität und die Geschwindigkeit des Responses, gleichzeitig unter parametrischer Unsicherheit garantiert. Strukturelle Fließbildalternativen, insbesondere die Verknüpfung von Reaktoren untereinander und Alternativen in Bezug auf die Regelungsstruktur, d.h. insbesondere die Kopplung von geregelten und manipulierten Variablen, zählen zu den Freiheitsgraden des Entwurfsprozesses. Des Weiteren werden unbenutzte Reaktoren und Regler im Netzwerk zugelassen, sodass a-priori keine feste Anzahl von benutzten Reaktoren und Reglern vorgegeben werden muss. Der optimale Entwurf des Reaktornetzwerks im offenen oder geschlossenen Regelkreis wird durch die Lösung eines einzelnen Optimierungsproblems ermittelt.

Der vorgeschlagene Ansatz erlaubt eine integrierte Behandlung von parametrischen Unsicherheiten, die entweder aus Modellunsicherheiten resultieren, wie z.B. Konstanten in der Reaktionskinetik oder Wärmeübergangskoeffizienten, oder aus Prozessunsicherheiten, die auch langsame Veränderungen des Zuflusses oder der Qualität der Edukte einschlieen. Es wird eine robuste Zwangsbedingung für die Eigenwerte formuliert, um ein robustes Verhalten des entworfenen Reaktornetzwerkes zu garantieren. Effiziente Formulierungen zur Verknüpfung von Reaktoren und neue Zwangsbedingungen zur Auswahl der Regelungsstruktur, die auf Komplementarität basieren, werden vorgeschlagen. Die Methode resultiert in einem semi-infiniten gemischt-ganzzahligen nichtlinearen Optimierungsproblem mit Komplementaritätsbdingungen, Disjunktionen, und einer robusten Eigenwert-Nebenbedingung. Es wird eine hybride zweistufige Lösungsmethode vorgeschlagen, welche die Lösungsalgorithmen des verwandten Optimierungsproblems integriert. Die vorgeschlagene Lösungsmethode wird auf eine Fallstudie der Allylchlorid-Produktion mit bis zu zehn Rohrreaktoren bzw. Rührkesselreaktoren angewandt.

https://doi.org/10.51202/9783186952035 - Generiert durch IP 216.73.216.36, am 18.01.2026, 21:11:22. © Urheberrechtlich geschützter Inhalt. Ohne gesonderte
Erlaubnis ist jede urheberrechtliche Nutzung untersagt, insbesondere die Nutzung des Inhalts im Zusammenhang mit, für oder in KI-Systemen, KI-Modellen oder Generativen Sprachmodellen.

1 Introduction

1.1 Reactor network synthesis

1.1.1 Motivation

Reactor network synthesis is a classical design problem in process systems engineering, which is defined as follows [19]: *"For given reaction stoichiometry, rate laws, a desired objective and system constraints, what is the optimal reactor network structure and its flow pattern? Where should mixing, heating, and cooling be introduced into the network?"* The essence of reactor network synthesis is to find the optimal reactor types, dimensions, operating conditions and the structural connections with respect to certain design objectives.

The reaction step is often the heart of a chemical process and the foundation for the process design. The reaction chemistry determines the character of the entire process and has a significant impact on its design [154]. The conversion rate of raw material, the operating cost and also the dynamic properties of a process are largely influenced by its reactors. For this reason, designing the reaction section of a flowsheet is one of the most important tasks of process design.

However, because of nonlinearity, uncertain rate laws, and typically a large number of reactor types and network structures, reactor network synthesis is one of the most challenging problems for process engineers. In this work, we will focus on this topic and propose a systemic method for open-loop and closed-loop reactor network synthesis. In this section, we first review different design methods for open-loop reactor network synthesis.

1.1.2 Task

The task of reactor network synthesis includes the following three subtasks: (1) Determination of the reactor network structure, including the number and type of each reactors; (2) determination of the operating points of each reactors; (3) determination of the design parameters, e.g., operating and equipment design parameters. Note that, although we have subdivided the task of reactor network synthesis into three subtasks, the decisions made in one subtask influence the decisions in the others.

The determination of the reactor network structure is to design the connection patterns of all reactors. Reactors can be arranged in parallel, in series or in a more complex manner. One has also to determine the feed patterns of all reactors, i.e., how each reactor is fed with raw materials. Raw materials may be fed into a single reactor, or into several reactors simultaneously. Besides, a reactor network may be allowed to include different types of reactors, e.g., CSTRs and PFRs, if better economical and operational performances can be achieved. In this case, the number and the type of reactors have to determined in this step. Recycles and bypasses may be also allowed. Hence, the determination of the reactor network structure is an essential and probably the most complicated task of reactor network synthesis.

https://doi.org/10.51202/9783186952035 - Generiert durch IP 216.73.216.36, am 18.01.2026, 21:11:22. © Urheberrechtlich geschützter Inhalt. Ohne gesonderte
Erlaubnis ist jede urheberrechtliche Nutzung untersagt, insbesondere die Nutzung des Inhalts im Zusammenhang mit, für oder in KI-Systemen, KI-Modellen oder Generativen Sprachmodellen.

The determination of the operating points of each reactor also fixes the steady state of each reactor, i.e., the component concentrations and the reactor temperature. Due to nonlinearity, a reactor may have different operating points when the same inlets and outlets are presented. However, both the steady-state performance and the dynamic properties of each reactor at a given operating point are not the same.

The determination of the design parameters is to specify, e.g., the working pressure of each reactor, the reactor dimensions, or the heating/cooling capacities of the reactor jacket. Note that the reactor network synthesis problem typically considers only the open-loop design problem. The design of a closed-loop control system for the reactor network is typically considered separately, refer to Section 1.3.

Task (1) can be referred as the design task of the entire network, while tasks (2) and (3) refer to the design task of individual reactors. The reactor network synthesis problem is much more complicated than designing a single reactor or a process flowsheet with fixed structure due to task (1), which introduces a large number of extra design degrees of freedom.

1.1.3 Methods

Methods for reactor network synthesis can be classified into heuristic methods, attainable region methods, superstructure methods and methods based on elementary process functions (EPF), refer to Fig. 1.1. A brief review of these methods can be found in [92, 133] for example and in standard textbooks for reactor design and reaction engineering, including [48, 141].

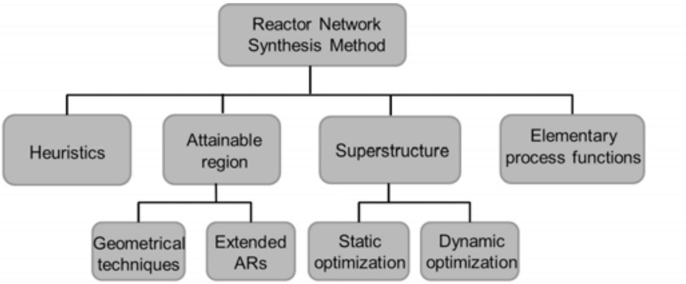


Figure 1.1: Reactor network synthesis methods.

Heuristic methods. Heuristic rules are often used to design chemical processes [35, 109, 110]. These rules are in general derived from reaction engineering knowledge. In some cases, heuristic methods do lead to good design results, e.g. [150]. But as the name “heuristic” indicates, these rules may not hold for any arbitrary reactor network and often they lead to suboptimal designs. Furthermore, if a process is of significant complexity, heuristic methods may become difficult or even impossible to be applied. For these reasons, research focuses on non-heuristic design methods, which are introduced in the following.

https://doi.org/10.51202/9783186952035 - Generiert durch IP 216.73.216.36, am 18.01.2026, 21:11:22. © Urheberrechtlich geschützter Inhalt. Ohne gesonderte Erlaubnis ist jede urheberrechtliche Nutzung untersagt, insbesondere die Nutzung des Inhalts im Zusammenhang mit, für oder in KI-Systemen, KI-Modellen oder Generativen Sprachmodellen.

Attainable region methods. Attainable region (AR) methods are also called targeting methods, because they aim at an achievable bound on the performance index of a system irrespective of the actual reactor configuration [92]. AR methods were first proposed in [70] and extended later in [39, 55, 68]. In classical AR methods, the attainable region is defined as a convex hull of concentrations, which can be achieved starting from the feed point by reaction and mixing. However, because a convex hull of the feasible region is derived graphically, the classical AR approaches can only be applied to synthesis problems which can be reduced to at most three dimensions [133]. Also, another difficulty in applying the AR methods is their integration with other synthesis methods for complete process design [133]. These properties restrict the application of AR methods. To overcome some of the difficulties in applying classical AR methods, Biegler and co-workers proposed hybrid methods [11, 12, 92, 147], which integrate the AR method with numerical optimization.

Superstructure approach. The superstructure approach for reactor network synthesis has been under investigation for more than 30 years. The approach is based on rigorous optimization. Depending on the type of the resulting optimization problem, the approach can be classified into superstructure approach using static or dynamic optimization. The first presentation of the superstructure approach can be dated back to [73]. Important extensions have been reported in [1, 2, 86, 88]. In this approach, a superstructure of a reactor network containing different candidates of connection patterns of the reactors is postulated first. Based on the usage of numerical optimization tools, an optimal reactor network which maximizes a given objective function is derived at the end.

The superstructure approach typically results in a complex mixed-integer nonlinear program (MINLP), e.g. [86, 88], or a mixed-integer dynamic optimization (MIDO) problem, e.g. [1, 2, 154]. These optimization problems are typically solved sequentially by appropriate numerical solvers. The key advantages of the superstructure approach are that it allows an arbitrarily general network, and still determines the optimal reactor network configuration and operating conditions [2]. However, this approach often leads to large MINLP problems, which are very hard to solve even locally. Furthermore, a significant limitation is that the optimal solution can only be as rich as the superstructure [154].

EPF methods. Elementary process function (EPF) methods [50, 133] have been proposed recently for the design of optimal chemical reactors. Its basic idea is to track a fluid element on its way through a reactor and optimize the reaction as well as mass transfer fluxes along its way. The methods consider the best reaction route in the thermodynamic state space. The optimal states of the fluid with respect to a certain objective function change along the reaction coordinates and they follow an optimal route in state space [133]. To optimize the chosen objective, reaction conditions must always be accomplished and the variables to control the states of the reacting fluid must also be changed along the path of the fluid element. A great advantage of this type of method is that no specific apparatus has to be assumed a priori. Hence, the method may suggest the development of innovative reactor concepts.

1.2 Design of decentralized control systems

A dynamic system can be controlled either by centralized or decentralized control systems. This classification stresses the point whether there is a single central controller or whether there are multiple decentralized controllers. Although decentralization can be

https://doi.org/10.51202/9783186952035 - Generiert durch IP 216.73.216.36, am 18.01.2026, 21:11:22. © Urheberrechtlich geschützter Inhalt. Ohne gesonderte Erlaubnis ist jede urheberrechtliche Nutzung untersagt, insbesondere die Nutzung des Inhalts im Zusammenhang mit, für oder in KI-Systemen, KI-Modellen oder Generativen Sprachmodellen.

carried out in different ways, throughout this work we will restrict the discussion to fully decentralized control systems with single-input single-output (SISO) proportional-integral (PI) controllers. Hence, decentralized PI control results in a multi-input multi-output (MIMO) system, in which each closed loop always couples a single CV and a single MV.

Decentralized PI control systems are most popular in industrial applications, because centralized control systems are expected to come with large complexity and design cost, leading to difficulties in implementation, tuning and maintenance [29]. Decentralized control is a favorable choice because of its low computational demand [10] and less measurements and information are needed to be transmitted [157]. These properties make decentralized control systems a preferred choice to control large and complex systems in practice.

Decentralized PI control systems will be assumed in this work for simultaneous design of a reactor network and its control system. Therefore, in this section we review the fundamentals of designing decentralized control systems for a given process, while the task of simultaneous process and control system design will be reviewed in the next section.

1.2.1 Tasks

The tasks of designing a decentralized control system include (1) the selection of candidate CV and MV, (2) the determination of the control structure, i.e., the pairing of candidate CV with MV, and (3) the tuning of control parameters. Note that, although we have subdivided the task of designing a decentralized control system into three subtasks, the decisions made in one subtask often influence the decisions in the others.

Selection of candidate CV and MV

Selection of candidate controlled variables (CV) and manipulated variables (MV) refers to the decisions regarding the number and type of candidate CV and MV. This first step of designing a decentralized control system answers the question which valves (candidate MV) can be manipulated and which quantities (candidate CV) can be controlled. This step is often based on an available process model and performed prior to the physical realization of the plant. A comprehensive review of how to select candidate CV and MV can be found in [176].

Candidate MV can be manipulated by controllers. They are physically identified as valves, which can be opened or closed during the operation. Here, “candidate” stresses the point that each single MV may or may not be manipulated in the final design. When a candidate MV is not manipulated in the final design, this MV is actually a design parameter and the corresponding valve position is fixed during process operation.

Candidate CV refer to quantities, which can be measured by sensors, including temperature, pressure and concentration, and controlled by a reference signal. The measured signal is fed to controllers, which in turn is used to manipulate valves. Likewise, “candidate” stresses here that each single candidate CV may or may not be measured and controlled in the final design. When a candidate CV is not used in the final design, the corresponding quantity does not need to be measured physically. Selection of candidate MV and CV results in two sets of variables, which refer to candidate MV and CV, respectively.

https://doi.org/10.51202/9783186952035 - Generiert durch IP 216.73.216.36, am 18.01.2026, 21:11:22. © Urheberrechtlich geschützter Inhalt. Ohne gesonderte Erlaubnis ist jede urheberrechtliche Nutzung untersagt, insbesondere die Nutzung des Inhalts im Zusammenhang mit, für oder in KI-Systemen, KI-Modellen oder Generativen Sprachmodellen.

Determination of control structure

The determination of the structure of a decentralized PI control system refers to the task of coupling individual candidate MV and CV, which are determined from the previous step. It specifies which of the MV is manipulated by which of the CV. Because each coupling has to be decided among multiple candidate MV and CV, this task leads to a combinatorial problem. Selecting the best alternative is therefore challenging, even for a relative small number of candidate MV and CV.

The determination of the control structure is critical because a poor decision may lead to fundamental performance limitations of the controlled system, which can not be overcome by controller turning. The decisions on pairing MV and CV is as important as controller design itself.

Despite its importance, limited attention has been paid to the design of control structures [175]. In practice, this task is often done based on physical understanding of the process. Therefore, there often exist a large number of possible pairings, which are not carefully considered during design. The combinatorial nature of selecting a control structure underlines the need for a systematic procedure for choosing the best control structure.

Controller tuning

Controller tuning refers to the task of determining the parameters of the controllers. For PI controllers, the parameters are the proportional and integral gains. The task of controller tuning sometimes refers to “controller design” in literature, which assumes that the control structure is known and fixed.

The ultimate goal of controller tuning is to realize desired dynamic closed-loop properties. Stability is a basic property, which must be ensured, but there exists other desired properties, including no overshoot, response speed, or settling time. The task of controller tuning is, however, not trivial, because the control parameters can not be related quantitatively and directly to time domain performance. Furthermore, controller tuning should be considered together with the determination of the control structure, because both tasks influence the dynamic properties of the closed-loop system.

1.2.2 Methods

Methods to design decentralized control systems can be classified into heuristic methods, indices-based methods and mathematical programming methods (cf. Fig. 1.2). Though the task of designing decentralized control systems can be subdivided into three subtasks, we will focus on only the last two subtasks, namely on control structure selection and controller tuning. The first subtask, selecting candidate CV and MV, is typically restricted by the available actuators and sensors in a given open-loop process.

Heuristic methods

Luyben et al. [109, 111] proposed a nine-step heuristic design procedure for plant-wide control design. The proposed procedure results in an effective plant-wide control structure for a given process flowsheet. Niederlinski [126] proposed a heuristic approach to the design problem of linear multivariable interacting control systems. The proposed heuristics are

https://doi.org/10.51202/9783186952035 - Generiert durch IP 216.73.216.36, am 18.01.2026, 21:11:22. © Urheberrechtlich geschützter Inhalt. Ohne gesonderte
Erlaubnis ist jede urheberrechtliche Nutzung untersagt, insbesondere die Nutzung des Inhalts im Zusammenhang mit, für oder in KI-Systemen, KI-Modellen oder Generativen Sprachmodellen.

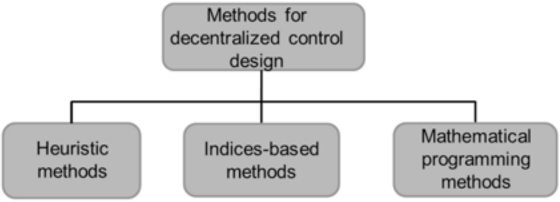

Methods for decentralized control design
Heuristic methods
Indices-based methods
Mathematical programming methods

Figure 1.2: Methods for the design of decentralized control systems.

used to make decisions on the control structure and the controller settings. Konda et al. [89] proposed an integrated framework of simulation and heuristics.

Indices-based methods

Indices-based methods rely on the use of an index, which is typically a real scalar derived from a process model, to evaluate and compare different control designs. This type of methods have mainly been developed for linear systems with a given operating point. According to the types of indices used, e.g. relative gain array (RGA) [24], Niederlinski index (NI) [126], singular values (SV) [85], or structured singular value (SSV) [93], indices-based methods can be further classified. RGA constitutes the first systematic index for interaction analysis and input-output pairing for linear multi-variable plants. It is still the most widely used technique in industry. RGA is based on the steady-state gain matrix of an open-loop plant, which can easily be obtained in practice by performing step tests. RGA has also been extended and generalized to other useful RGA-based derivatives, e.g., to the dynamic gain array (DRGA) [184], or the generalized relative dynamic gain [51]. NI is similar to RGA and does actually not provide more information for control structure selection [155]. The relationship between RGA and NI is explored in [29]. RGA and NI were initially developed for stable processes, but they have also been extended for input-output pairing of unstable MIMO systems [71]. A comprehensive review of RGA-based methods is given in monograph [83].

Methods based on mathematical programming

Due to advances in computational techniques and due to increased computing power, control design methods based on mathematical programming [28, 62, 81, 90, 114, 117, 124] emerge. In this class of methods, the control design problem is formulated as an optimization problem, e.g., a mixed-integer linear program (MILP), a mixed-integer nonlinear program (MINLP) or a mixed-integer dynamic optimization (MIDO) problem. The formulated optimization problems are solved by appropriate numerical solvers, which generate the optimal control design. The criteria guiding control design can either be based on the trajectories of the closed-loop system [28, 62, 117, 124], or certain indices to assure controllability (e.g. RGA) [28, 81, 90, 114]. The selected criteria are either the objective

https://doi.org/10.51202/9783186952035 - Generiert durch IP 216.73.216.36, am 18.01.2026, 21:11:22. © Urheberrechtlich geschützter Inhalt. Ohne gesonderte
Erlaubnis ist jede urheberrechtliche Nutzung untersagt, insbesondere die Nutzung des Inhalts im Zusammenhang mit, für oder in KI-Systemen, KI-Modellen oder Generativen Sprachmodellen.

function to be minimized/maximized or they are required to take specific values, or to be bounded.

Here we stress the formulation for control structure selection proposed by Narraway and Perkins [124], on which many later works [28, 90, 117], as well as this work, are based. In this formulation [124], decisions for pairing CV and MV are modeled by integer variables to represent how single CV and MV are coupled. The formulation has been successfully applied to some case studies. However, a major drawback of this formulation is that the generated mixed-integer optimization problem contains a large number of integers and is computationally demanding. In this work, we will propose an alternative formulation, which is based on complementarity (cf. Section 4.1.2). The proposed formulation can be treated more efficiently by numerical solvers.

A great advantage of the methods based on mathematical programming is that they enable a straightforward integration of process and control system design (cf. Section 1.3). A single optimization problem is typically formulated and solved at once, which determines both the process and control system design parameters, cf. [28, 62, 117, 124].

1.3 Simultaneous process and control system design

Process and control system design are typically two separate sequential steps in designing closed-loop processes. The goal of process design is to select a most economical process flowsheet and an operating condition, while the goal of control design is to ensure desired dynamic properties by selecting a control structure and its parameters. Because the dynamic properties obtained in the second step (control design) depend on the design results from the first step (process design), this two-step procedure may lead to suboptimal closed-loop solutions. Simultaneous process and control system design tries to treat these problems simultaneously by integrating these two sequential steps into a single step.

1.3.1 Motivation

Process flowsheets are typically designed in two sequential steps: first the process design is fixed and then the control system is added. In the first step (process design), an economical cost function is optimized subject to a steady-state open-loop process model. Dynamic properties such as stability and operability are typically not considered in this step and can hence not be guaranteed. In the second step (control system design), the control system is designed and closed-loop dynamic properties are taken into consideration. Because this procedure ignores the interrelation between process and control system design, it may result in an open-loop design which is difficult to control, and may hence results in an unsatisfactory closed-loop design.

The motivation of simultaneous process and control design is to integrate these two sequential steps. Both, static performance, e.g., economics, and dynamic performance, e.g., stability and set-point tracking, are ensured in a single step. Process design decisions, including flowsheet structure, equipment parameters, or operating conditions, and process control decisions, including control structure or controller tuning, will be considered in an integrated framework. Systematic methods for simultaneous process and control design will be briefly reviewed next.

https://doi.org/10.51202/9783186952035 - Generiert durch IP 216.73.216.36, am 18.01.2026, 21:11:22. © Urheberrechtlich geschützter Inhalt. Ohne gesonderte Erlaubnis ist jede urheberrechtliche Nutzung untersagt, insbesondere die Nutzung des Inhalts im Zusammenhang mit, für oder in KI-Systemen, KI-Modellen oder Generativen Sprachmodellen.

1.3.2 Methods

Simultaneous process and control design is not a new topic; there rather exist already hundreds of publications in this field. For a comprehensive overview of this topic, we refer to recent review papers [144, 148, 156, 179, 188]. According to [179], methods for simultaneous process and control system design can be classified in two ways: approaches which systematically examine the dynamic properties of alternative designs, so-called projecting methods, and approaches which perform process and control design at once by solving an optimization problem, so-called integrated optimization methods (cf. Fig. 1.3). Since interrelated issues are often considered, there is no strict and agreed classification.

Projecting methods predict and compare process dynamics by means of controllability indices for candidate design alternatives during the design phase. This class of methods belong to the earliest methodologies reported in literature, which explicate and treat the conflicts between process and control design. Projecting methods can be further classified into methods based on input-output controllability, methods based on state controllability, process-oriented methods, methods based on steady-state multiplicity and methods based on phenomenological models [179]. For methods based on input-output or state controllability, different process alternatives are studied by optimizing a steady-state process model economically with the consideration of open-loop controllability indices. Controllability indices used here are mainly focused on the effects of perturbations to the operating constraints and their propagations through the process flowsheet [179]. Process-oriented methods consider the task of simultaneous process and control design for specific processes. Methods based on multiplicity come from steady-state multiplicity analysis and focus on integrating operability with reactor design. Phenomenological methods apply phenomenological knowledge of the process to distinguish the designs with best dynamic performance, using sensibility analysis of the thermodynamic properties of the chemical process or specifically passivity theory [179].

The integrated optimization methods introduce dynamic performance measures and use them to formulate a comprehensive optimization problem for the determination of the best economical and controllable plant including the design of a control system [179]. Integrated optimization methods can be further characterized by the scope of design problems, techniques to quantify dynamic performance, control strategies, the treatment of perturbations/uncertainties and the types of the optimization problems formulated [179] (cf. Fig. 1.3). The derived optimization problems often consider different subsets of design decisions, e.g., operating point, design parameters, flowsheet structure, controller types and control structure.

Note that, although a number of methods exist in literature for simultaneous process and control design, numerous issues still remain open for research. This is mainly due to the joint and integrated nature of process and control design, in which many potential possibilities and extensions considering different process/control design aspects exist. With the advance of computational techniques, the reliable solution of the resulting numerical problems becomes more and more mature.

https://doi.org/10.51202/9783186952035 - Generiert durch IP 216.73.216.36, am 18.01.2026, 21:11:22. © Urheberrechtlich geschützter Inhalt. Ohne gesonderte Erlaubnis ist jede urheberrechtliche Nutzung untersagt, insbesondere die Nutzung des Inhalts im Zusammenhang mit, für oder in KI-Systemen, KI-Modellen oder Generativen Sprachmodellen.

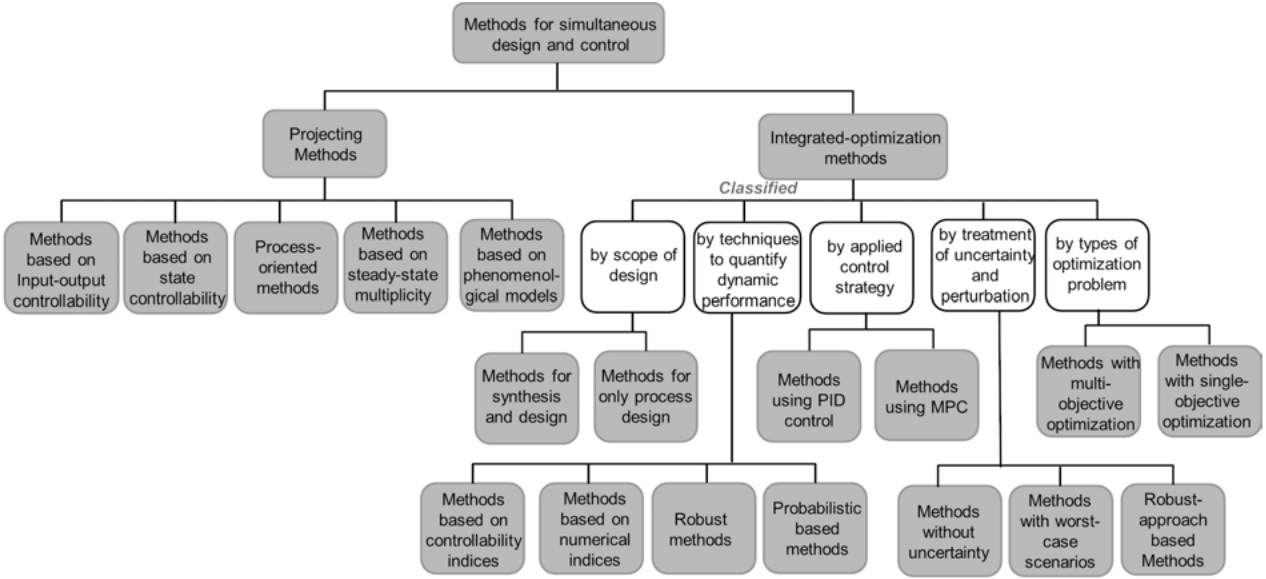

Methods for simultaneous design and control
Projecting Methods
Integrated-optimization methods
Classified
Methods based on Input-output controllability
Methods based on state controllability
Process-oriented methods
Methods based on steady-state multiplicity
Methods based on phenomenol-gical models
by scope of design
by techniques to quantify dynamic performance
by applied control strategy
by treatment of uncertainty and perturbation
by types of optimization problem
Methods for synthesis and design
Methods for only process design
Methods using PID control
Methods using MPC
Methods with multi-objective optimization
Methods with single-objective optimization
Methods based on controllability indices
Methods based on numerical indices
Robust methods
Probabilistic based methods
Methods without uncertainty
Methods with worst-case scenarios
Robust-approach based Methods

Figure 1.3: Methods for simultaneous design and control [179].

https://doi.org/10.51202/9783186952035 - Generiert durch IP 216.73.216.36, am 18.01.2026, 21:11:22. © Urheberrechtlich geschützter Inhalt. Ohne gesonderte
Erlaubnis ist jede urheberrechtliche Nutzung untersagt, insbesondere die Nutzung des Inhalts im Zusammenhang mit, für oder in KI-Systemen, KI-Modellen oder Generativen Sprachmodellen.

1.4 Content and goals of this work

In the above sections we have reviewed three topics, which relate closely to this work. We were not able to cover every aspect, but based on the presented reviews, we have already glimpsed at the motivation of this work. In this section, we will present a thorough overview of this work, focusing on the new features and on its relation to others in literature.

1.4.1 Overview

This work presents a systematic approach to simultaneously design open- or closed-loop reactor networks with robust dynamic properties. Our goal is to find a process structure and a steady-state operating point, in either open- or closed-loop, which is not only economically optimal but also possesses two eigenvalue-based dynamic properties, namely stability and a specified response speed. Both, alternative flowsheet structures, i.e., how different reactors are connected with each other, and the alternative PI control structure, i.e., different pairings of CV and MV (cf. Fig. 1.4), are simultaneously considered and optimized. Process design parameters, i.e., reactor size, and control parameters, i.e., proportional control gains, are determined through solving a single optimization problem. Parametric uncertainties, which may either refer to model uncertainties, e.g., reaction kinetic constants or heat transfer coefficients, or to process uncertainties, e.g., slow disturbances in load or quality of raw materials, are also considered in this approach. By considering uncertainty, it is guaranteed that the designed process is robust with respect to the specified dynamic properties. Not only the nominal operating point of the designed process, but also the operating points close to the nominal operating point are stable and/or have specified response speed. The approach results in a robust steady-state process design, either open- or closed-loop, which is economically optimal and has desired dynamic properties. We note that, the proposed method has been developed for reactor networks but it carries over to other (integrated) process and/or control synthesis problems. The presented formulation becomes much simpler, if no/less structural alternatives are considered. Some important features of this work will be discussed below in more detail.

Modeling of reactor networks

A structured modeling procedure for reactor network synthesis with the consideration of both flowsheet and control structural alternatives is presented. The procedure results in a compact dynamic model of reactor networks for open-loop and closed-loop design. The open-loop model considers structural flowsheet alternatives, while the closed-loop model for simultaneous process and control design considers control structure alternatives in addition.

The proposed modeling procedure is based on a superstructure approach [19], but has the following two new features: First, the procedure leads to a compact ODE model, which has internal mathematical structure referring to the flow connections between reactors. This compact ODE model, especially its internal mathematical connectivity structure, is important for the analysis and the synthesis of eigenvalue-based dynamic properties of the designed process. Second, the superstructure approach, which is typically used for open-loop process synthesis, is extended to simultaneously treat control structure alternatives.

https://doi.org/10.51202/9783186952035 - Generiert durch IP 216.73.216.36, am 18.01.2026, 21:11:22. © Urheberrechtlich geschützter Inhalt. Ohne gesonderte Erlaubnis ist jede urheberrechtliche Nutzung untersagt, insbesondere die Nutzung des Inhalts im Zusammenhang mit, für oder in KI-Systemen, KI-Modellen oder Generativen Sprachmodellen.

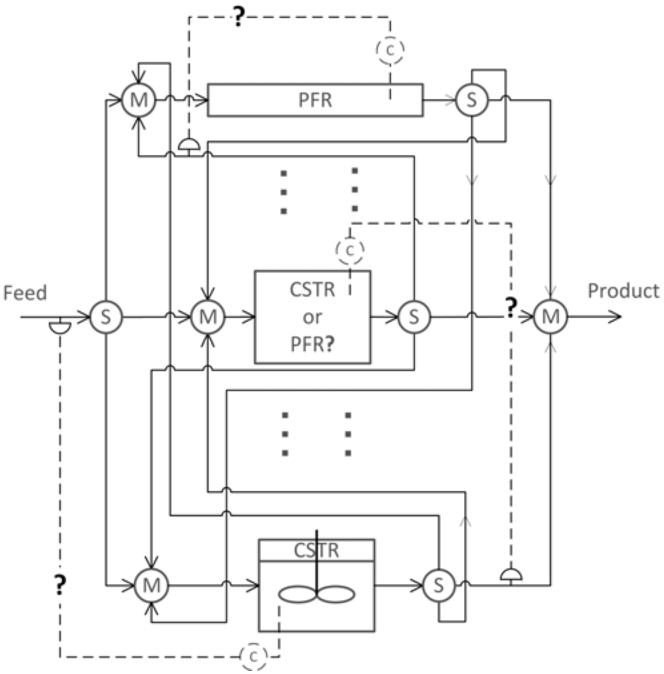


Figure 1.4: A closed-loop reactor network superstructure with undetermined process and control structures for simultaneous process and control design. The figure illustrates the design problem considered in this work. CSTR refers to a continuous stirred-tank reactor, PFR to a plug flow reactor. M and S refer to mixers and splitters, respectively. C stands for a PI controller. Solid arrows represent candidate flow connections, while dashed lines represent candidate couplings between MV and CV. Both, the flowsheet and control structures are not fixed a priori.

The resulting closed-loop model can therefore handle both flowsheet and control structure alternatives as degrees of freedom in the design.

To model open-loop reactor networks, we used the reactor network superstructure presented in [87] as a starting point for further development. This superstructure comprises a fundamental setting, to which we will stick throughout this work. The superstructure is similar to the one in Fig. 1.4, except that no control loops were included in [87]. The superstructure consists of several reactors (either CSTR or PFR), which are connected through candidate flow connections. Depending on the existence of any flow connection in the final design, reactors can be connected in different ways. For example, reactors in series or in parallel are two simple flowsheet structures, while there exist more complicated connection patterns. We want to stress an important modeling trick proposed in this work to model structural alternatives: The outlets of a reactor can be modeled by the product

https://doi.org/10.51202/9783186952035 - Generiert durch IP 216.73.216.36, am 18.01.2026, 21:11:22. © Urheberrechtlich geschützter Inhalt. Ohne gesonderte
Erlaubnis ist jede urheberrechtliche Nutzung untersagt, insbesondere die Nutzung des Inhalts im Zusammenhang mit, für oder in KI-Systemen, KI-Modellen oder Generativen Sprachmodellen.

of a corresponding flowrate variable with a vector of component concentrations and energy densities. By using this trick, one can distinguish between existent and non-existent flow connections just by checking the values of the flowrate variables. This results in a dynamic model of reactor networks, which has an internal mathematical structure for connectivity without using integer variables. This trick also plays a central role in the eigenvalue-based analysis of the derived model and influences the proposed reformulations.

The closed-loop reactor network model for simultaneous process and control design is obtained by extending the open-loop model by additional decentralized PI control loops. The basic problem setting of designing the structure of the decentralized PI control system is taken from [124]. This work extends their work by simultaneously considering flowsheet alternatives and proposes an efficient complementarity-based formulation for control structure selection. Candidate CV and MV are first defined by inspecting the open-loop model; subsequently they are paired to form candidate decentralized control loops. The modeling procedure of closed-loop reactor networks results in a dynamic model, in which both flowsheet and control structure alternatives are degrees of freedom of the design. Eigenvalue-based analysis of the dynamic properties of the closed-loop model is carried on in a similar way as it is done for the open-loop reactor network model.

Robust eigenvalue constraints for dynamic properties

Another feature of this work is to ensure dynamic properties, namely stability and specified response speed, of the final design in either open- or closed-loop. This is achieved by imposing so-called eigenvalue constraints. The considered dynamic properties are formulated by using the eigenvalues of the Jacobian matrix of the reactor network model, or more precisely, specifying an upper bound on the spectral abscissa (SA) of the system's Jacobian matrix. From nonlinear systems theory, we know that SA determines stability and response speed (cf. Chapter 2). When SA is less than zero, a dynamic system is locally stable; the more negative the SA, the faster a dynamic system responds to disturbances. Therefore, by constraining the SA of a reactor network, stability and a specified response speed in the final design can be ensured.

Parametric uncertainties are also taken into consideration. The type of parametric uncertainty is adopted from [119], where it is assumed that uncertain parameters lie in a certain uncertainty region around their nominal values. Input variables, disturbances and reference signals are assumed to vary quasi-statically compared to the system dynamics such that they can be also modeled in this way. The resulting robust eigenvalue constraint guarantees that not only the nominal design, but also nearby designs, have the desired dynamic properties.

Idle reactors and controllers

Considering idle reactors and controllers in the reactor network design problem is another important feature of this work. The motivation to do this is to select the optimal number and types of reactors and controllers in the final design, so that designers do not have to define a fixed number of reactors and controllers a priori. Idle reactors (idle controllers) refer to reactors (controllers) in the final design, which will not be physically implemented. A reactor is idle, if it has no flow connections with other reactors, while a PI controller is idle, if it is not involved in a closed loop. In the proposed model, idle reactors and

https://doi.org/10.51202/9783186952035 - Generiert durch IP 216.73.216.36, am 18.01.2026, 21:11:22. © Urheberrechtlich geschützter Inhalt. Ohne gesonderte Erlaubnis ist jede urheberrechtliche Nutzung untersagt, insbesondere die Nutzung des Inhalts im Zusammenhang mit, für oder in KI-Systemen, KI-Modellen oder Generativen Sprachmodellen.

controllers can be identified just by checking the values of flowrate variables and controller gain parameters, respectively. So, one can start with a general superstructure, which contains a sufficient number of reactors and controllers, and let the optimizer decide how many and which reactors and controllers are idle in the final design.

The consideration of idle reactors and controllers makes the formulation and the analysis of eigenvalue constraints for reactor network synthesis much more complicated, because eigenvalue-based dynamic properties can not be formulated straightforwardly for the dynamic model of the superstructure by applying mathematical nonlinear systems theory. Because only non-idle reactors and controllers will be implemented physically, the submodels of idle/non-idle reactors and controllers must be distinguished and considered separately. This results in four submodels for the closed-loop reactor network model, i.e., a submodel for idle reactors, a submodel for idle controllers, a submodel for non-idle reactors and a submodel for non-idle controllers. The Jacobian matrix of the total model also has an internal structure corresponding to the submodels. Since we are only interested in the dynamic properties of non-idle reactors and controllers, a novel formulation of an eigenvalue constraint is introduced, which refers only to the eigenvalues of non-idle reactors and controllers.

Integer (binary) variables are introduced to represent the status (idle or non-idle) of each reactor and controller. The proposed eigenvalue constraint for non-idle reactors and controllers is found to be discontinuous. The discontinuity is caused by activation/deactivation of idle reactors and controllers, which leads to a dimensional change of the eigenvalue spectrum of non-idle reactors and controllers. In order to use off-the-shell numerical toolboxes, we transform the proposed eigenvalue constraint from a discontinuous into a continuous function, which is smooth almost everywhere, by introducing integer variables. After this transformation, the resulting mixed-integer eigenvalue constraint is smooth with respect to its arguments almost everywhere and it can be treated by mixed-integer mathematical programming and eigenvalue optimization (cf. Section 5). We find that the established synthesis problem [87], where all reactors and controllers are assumed to be non-idle, is a rather simple special case, for which no integer variables are needed to derive a continuous eigenvalue constraint for reactor network design.

Problem formulation and solution method

In this work, we present two problem formulations. One is for open-loop reactor network synthesis, and the other is for simultaneous reactor network and control design. The open-loop design problem results in a semi-infinite MINLP with a robust eigenvalue constraint, while the simultaneous closed-loop design problem constitutes a semi-infinite MINLP with a robust eigenvalue constraint and additional complementarity constraints and disjunctions. In both formulations, a nonlinear economic objective function is minimized, which is subject to the steady-state process model either in open- or in closed-loop and a robust eigenvalue constraint for dynamic properties. Integers represent the existence of a reactor and a controller in the final design. The number of introduced integers equals the number of reactors and controllers included in the superstructure. Complementarity constraints appear in the simultaneous closed-loop design problem referring to the constraints for control structure selection, i.e., the different pairings of candidate CV and MV.

The formulated problems are challenging to solve, because they combine features coming from mixed-integer nonlinear programs (MINLP), mathematical programs with comple-

https://doi.org/10.51202/9783186952035 - Generiert durch IP 216.73.216.36, am 18.01.2026, 21:11:22. © Urheberrechtlich geschützter Inhalt. Ohne gesonderte
Erlaubnis ist jede urheberrechtliche Nutzung untersagt, insbesondere die Nutzung des Inhalts im Zusammenhang mit, für oder in KI-Systemen, KI-Modellen oder Generativen Sprachmodellen.

mentarity constraints (MPCC), semi-infinite optimization (SIP) and eigenvalue optimization (EVO) (cf. Section 5). To the author's knowledge, the most challenging feature is the treatment of robust eigenvalue constraints, which is rarely discussed in literature. Eigenvalue constraints are in general non-smooth. Hence, we need to find the global minimum of a non-smooth function. Besides, one has to take care of integer variables and many complementary constraints and disjunctions, which makes the solution task challenging both theoretically and practically.

As a first pragmatic approach, we propose a two-step hybrid solution strategy to solve the proposed optimization problem locally. The solution strategy solves first a deterministic problem without considering uncertainties (step 1) and afterwards solves a semi-infinite (uncertain) problem (step 2). In the first step, assuming smoothness of the eigenvalue constraints near the local optimum or using smoothing techniques for eigenvalue constraints (cf. Section 5.4), smooth optimization solvers, e.g., SNOPT [54], can be directly applied to solve the derived deterministic problem. The applied smooth solvers can be multiply initialized such that the global minimum can be approximated. In the second step, an uncertain problem is derived by fixing the integer variables to the results of the first step. The resulting uncertain problem is solved by applying the normal vector approach [120]. Advantage of this two-step hybrid solution strategy is that, the normal vector approach in the second step can be properly initialized using the solutions of the first step. If the global optimum can be approximately obtained in the first step (e.g., by using a multi start strategy), the second step often leads to good local optimal solutions.

1.4.2 New features of this work

Having presented an overview of the content and goals of this work, the four new features of this work are highlighted in Fig. 1.5. These features make this work an original contribution to the literature. Although there may exist other works which consider some of these features, to the author's knowledge nobody has considered these features in an integrated framework.

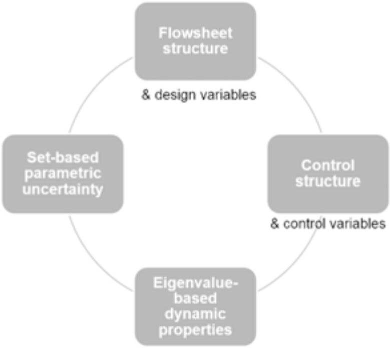


Figure 1.5: Four features in setting up the design problem of this work.

https://doi.org/10.51202/9783186952035 - Generiert durch IP 216.73.216.36, am 18.01.2026, 21:11:22. © Urheberrechtlich geschützter Inhalt. Ohne gesonderte Erlaubnis ist jede urheberrechtliche Nutzung untersagt, insbesondere die Nutzung des Inhalts im Zusammenhang mit, für oder in KI-Systemen, KI-Modellen oder Generativen Sprachmodellen.

Flowsheet and control structure alternatives are considered simultaneously in this work. Flowsheet structure alternatives refer to different connection patterns of the reactors in a reactor network, and control structure alternatives refer to different pairings of candidate CV and MV to form a decentralized PI control system. Established methods in literature typically consider only flowsheet structure alternatives without a control system, or only control structure alternatives for fixed flowsheet structures and process designs. In this work, however, we will consider the integrated case, in which both, the flowsheet and the control system are designed simultaneously.

Eigenvalue-based dynamic properties are guaranteed in the final design. In contrast to other characteristics which evaluate the performance of closed-loop systems, e.g., failure tolerance, set-point tracking, or dead time, the spectral abscissa is selected as a design criterion in this work. The spectral abscissa determines the stability and response speed of a designed system in either open- or closed-loop. It results in an eigenvalue constraint for the reactor network design problem. Idle reactors and controllers are considered in formulating the eigenvalue constraint to guarantee dynamic properties. The consideration of idle reactors and controllers makes it possible to determine the optimal number/type of reactors and controllers in the final design. Hence, dynamic properties (stability and fast response speed) of the final open-loop or closed-loop design are guaranteed by imposing eigenvalue constraints.

Parametric uncertainty is considered in formulating the eigenvalue constraint, to result in a robust design. In this work, parametric uncertainty is formulated in a non-probability setting through a hyper-rectangular uncertainty region, which is assumed to be known a priori. Although parametric uncertainty can be described in other ways, hyper-rectangular regions are often used in literature for simplicity [9, 117, 120, 146]. Uncertain parameters result in a robust eigenvalue constraint, which is challenging to be treated by numerical solvers. It guarantees that the final open-loop or closed-loop design has the desired dynamic properties in the uncertainty region around the normal operating point.

Relation to literature

Although there exist many papers which are relevant to the contents of this work, three papers [87, 120, 124] are directly related.

The basic problem settings of (open-loop) reactor network synthesis are adopted from [87]. The reactor network superstructure presented in [87] is directly used throughout this work. In comparison to [87], however, several advances have been achieved. First, idle reactors in open-loop superstructures are allowed and parametric uncertainties appear in the problem formulation. Second, instead of treating eigenvalue constraints by a conservative approximation, this work evaluates eigenvalue constraints exactly or uses specialized smoothing methods to approximate eigenvalue constraints (cf. Section 5.6). A more exact treatment of eigenvalue constraints leads to more exact approximations of the feasible region. Third, plug flow reactors are included in the reactor network superstructure, which allows an automatic decision on the use of different types of reactors.

The problem setting for designing decentralized control structures introduced in [124] is adopted in this work. Narraway and Perkins [124] proposed an integer-based formulation for control structure selection, which is adopted by many later works [28, 90, 117]. In this work, however, we address the same problem of selecting control structure, but present an alternative equivalent formulation. The presented formulation is based on complementarity

https://doi.org/10.51202/9783186952035 - Generiert durch IP 216.73.216.36, am 18.01.2026, 21:11:22. © Urheberrechtlich geschützter Inhalt. Ohne gesonderte Erlaubnis ist jede urheberrechtliche Nutzung untersagt, insbesondere die Nutzung des Inhalts im Zusammenhang mit, für oder in KI-Systemen, KI-Modellen oder Generativen Sprachmodellen.

constraints and can be treated by numerical optimization methods much more efficiently. Moreover, this work extends the work of [124] in the sense that both flowsheet and control structure alternatives are decided simultaneously.

Last but not least, the work of [120] has influenced this work. The formulation of parametric uncertainty and the usage of eigenvalue-based criteria are directly adopted from there. Also, in the proposed two-step solution strategy (cf. Section 5.6), we directly use the normal vector approach proposed in [120] to solve the resulting uncertain problem in the second step. However, this work differs from [120] because it considers flowsheet and control structure alternatives. The modeling of reactor networks and the analysis of eigenvalue-based properties in both open- and closed-loops are also independent from [120]. Furthermore, the first step of the proposed two-step solution strategy presents an alternative way to initialize the normal vector approach (cf. Section 5.6).

The content of our publications [189, 190] are re-used in this thesis. Major parts of Chapter 3 and the open-loop case study shown in Section 6.1 are reproduced from [189]. Chapter 4 and the closed-loop case study shown in Section 6.2 are reproduced from [190]. The proposed two-step solution method presented in Section 5.6 has originally been proposed in [190].

https://doi.org/10.51202/9783186952035 - Generiert durch IP 216.73.216.36, am 18.01.2026, 21:11:22. © Urheberrechtlich geschützter Inhalt. Ohne gesonderte
Erlaubnis ist jede urheberrechtliche Nutzung untersagt, insbesondere die Nutzung des Inhalts im Zusammenhang mit, für oder in KI-Systemen, KI-Modellen oder Generativen Sprachmodellen.

2 Some Preliminaries

This chapter introduces some theoretical concepts related to dynamic systems. Fundamental results on dynamic systems will be introduced first. After that we discuss eigenvalue and spectral abscissa functions. Lyapunov stability and response speed of dynamic systems will be analyzed by the spectral abscissa function. The extension of the presented results to differential-algebraic systems will be presented at the end.

2.1 Dynamic systems

Consider a nonlinear autonomous dynamic system represented by a set of ordinary differential equations (ODE),

$$\dot{x} = f(x),\ x(0) = x_0, \tag{2.1}$$

where $x(t) \in \mathbb{R}^m$ are the differential variables, or states, of the system. $f : \mathbb{R}^m \to \mathbb{R}^m$ is a general, not necessarily smooth, function. $x_0 \in \mathbb{R}^m$ denotes the initial condition. $\dot{x} = dx(t)/dt$ denote the derivatives of states $x(t)$ with respect to time t. System (2.1) is called autonomous, because $f(\cdot)$ is not explicitly a function of t.

$x(t)$ is a solution of the initial value problem (2.1) on interval $I = [0, t_1)$, $t_1 > 0$, if $x(t)$ is differentiable and

$$\begin{aligned} &\frac{dx(t)}{dt} \equiv f(x(t)), \forall t \in I, \\ &x(0) = x_0. \end{aligned}$$

Existence and uniqueness of solutions $x(t)$ of Eq. (2.1) relate to two questions, which should be addressed before looking for analytical or numerical solutions. It is known that these properties can be ensured by imposing some conditions on the function $f(x)$ [84]: If $f(x)$ is continuous, solutions of system (2.1) always exist and they are continuously differentiable. Uniqueness of solutions can be guaranteed by imposing Lipschitz continuity on $f(x)$, which is defined as follows:

Definition 2.1.1 (Local Lipschitz continuity)**.** *A function* $f : \mathbb{R}^m \to \mathbb{R}^n$ *is called locally Lipschitz continuous at a point* x^**, if there exist a neighborhood* U_{x^*} *of* $x^* \in \mathbb{R}^m$ *and a scalar* $L_0 > 0$*, so that* $\forall x_1, x_2 \in U_{x_0}$

$$\|f(x_1) - f(x_2)\| \leq L_0 \|x_1 - x_2\|. \tag{2.2}$$

A function $f(x)$ is said to be locally Lipschitz on an open subset $D \subseteq \mathbb{R}^m$, if $f(x)$ is locally Lipschitz at each individual point in D. Hence, L_0 in Eq. (2.2) may not be the same for different x^*. In contrary, global Lipschitz condition assumes that there is a uniform $L > 0$, which does not depend on the specific reference point.

Definition 2.1.2 (Global Lipschitz continuity)**.** *A function* $f(x)$ *is said to be globally Lipschitz continuous on an open subset* $D \subseteq \mathbb{R}^m$*, if there exists a (uniform)* $L > 0$*, so that*

$$\|f(x_1) - f(x_2)\| \leq L \|x_1 - x_2\|, \forall x_1, x_2 \in D.$$

https://doi.org/10.51202/9783186952035 - Generiert durch IP 216.73.216.36, am 18.01.2026, 21:11:22. © Urheberrechtlich geschützter Inhalt. Ohne gesonderte Erlaubnis ist jede urheberrechtliche Nutzung untersagt, insbesondere die Nutzung des Inhalts im Zusammenhang mit, für oder in KI-Systemen, KI-Modellen oder Generativen Sprachmodellen.

Local/global Lipschitz continuity has a close relationship to continuity and first order derivatives of $f(x)$:

Lemma 2.1.1 (Local Lipschitz continuity, Lemma 2.3 in [84]). *Let* $f(x)$ *be continuous on some domain* $D \subseteq \mathbb{R}^m$, *if* $\frac{\partial f}{\partial x}$ *exists and is continuous in* D, *then* f *is locally Lipschitz on* D.

Lemma 2.1.2 (Global Lipschitz continuity, Lemma 2.4 in [84]). *Let* $f(x)$ *be continuous on* $\mathbb{R}^m$. *If* $\frac{\partial f}{\partial x}$ *exists and is continuous on* $\mathbb{R}^m$, *then* f *is globally Lipschitz in* $\mathbb{R}^m$, *if and only if* $\exists K > 0$, *so that*

$$\|\frac{\partial f}{\partial x}(x)\| < K, \forall x \in \mathbb{R}^m.$$

That is, $\partial f / \partial x$ *is uniformly bounded on* $\mathbb{R}^m$.

Existence and uniqueness of solutions of system (2.1) can be guaranteed by the following theorems.

Theorem 2.1.3 (Local existence and uniqueness, Theorem 2.2 in [84]). *If* $f(x)$ *is locally Lipschitz continuous at an initial point* x_0, *then there exists a* $t_1 > 0$, *so that the initial value problem (2.1) has a unique solution* $x(t)$ *for* $t \in [0, t_1]$.

This theorem does not guarantee the existence of a solution $x(t)$ for arbitrary t_1. For example,

$$\dot{x} = -x^2, x(0) = -1,$$

has locally a unique solution $x(t) = (t-1)^{-1}$. But as $t \to 1$, $x(t) \to \infty$. So we can not find solutions for $t_1 \geq 1$.

Global existence and uniqueness of the solutions can be ensured by imposing stronger conditions. The following theorem establishes the existence of a unique solution for arbitrarily large t_1.

Theorem 2.1.4 (Global existence and uniqueness, Theorem 2.3 in [84]). *If* $f(x)$ *is globally Lipschitz continuous on* $\mathbb{R}^m$, *then the initial value problem (2.1) has a unique solution* $x(t)$ *for* $t \in [0, t_1]$, $\forall t_1 > 0$.

Theorem 2.1.4 is strong because it is based on the simple concept of global Lipschitz continuity. However, many dynamic system arising from engineering applications are modeled by a $f(\cdot)$, which is not globally Lipschitz continuous. Furthermore, one can also easily construct smooth problems, which are not globally Lipschitz, but which do have a unique global solution. For this reason, less restrictive conditions are of interest.

Local Lipschitz continuity of a function is basically a requirement for smoothness, which is implied by continuous differentiability. If we assume that models of physical systems are locally Lipschitz continuous, the following theorem results in global existence and uniqueness by imposing some additional properties of the solutions.

Theorem 2.1.5 (Global existence and uniqueness, Theorem 2.4 in [84]). *Let* $f(x)$ *be locally Lipschitz continuous in a domain* $D \subset \mathbb{R}^m$. *Let* W *be a compact subset of* D, $x_0 \in W$, *and suppose it is known that every solution of (2.1) lies entirely in* W. *Then, there is a unique solution of system (2.1) which is defined for* $t \in [0, t_1]$, $\forall t_1 > 0$.

Throughout this work, we assume that such a W always exists. Hence, system (2.1) has a unique solution $x(t)$ starting from x_0 for $t \in [0, +\infty)$, if the conditions of Lemma 2.1.1 hold.

https://doi.org/10.51202/9783186952035 - Generiert durch IP 216.73.216.36, am 18.01.2026, 21:11:22. © Urheberrechtlich geschützter Inhalt. Ohne gesonderte Erlaubnis ist jede urheberrechtliche Nutzung untersagt, insbesondere die Nutzung des Inhalts im Zusammenhang mit, für oder in KI-Systemen, KI-Modellen oder Generativen Sprachmodellen.

2.2 Eigenvalue and spectral abscissa functions

Let $\mathcal{M}_n$ be a vector space of $n \times n$ real matrices, which is not necessarily symmetric. Elements of this vector space are real square matrices. The zero vector of this vector space is the zero matrix and the vector space $\mathcal{M}_n$ is of dimension n^2. Equipped with any matrix norm, $\mathcal{M}_n$ is a metric space. Therefore, smoothness of matrix-valued functions can be defined.

Let $M : \mathbb{R}^m \to \mathcal{M}_n$ be a matrix-valued function. For $x \in \mathbb{R}^m$, $M(x) \in \mathcal{M}_n$ with elements $M_{i,j}(x)$, $i = 1, \cdots, n$, $j = 1, \cdots, n$. If we assume that $M_{i,j} : \mathbb{R}^m \to \mathbb{R}$ are smooth functions, the smoothness of function $M(\cdot)$ follows from the smoothness of functions $M_{i,j}(\cdot)$. In this work, we always consider the case where $M_{i,j}(\cdot)$ are smooth functions of x.

The eigenvalues of $M(x)$ are the roots of the n-th order polynomial

$$p_{M(x)}(\lambda) := det(M(x) - \lambda I),$$

where I is an $n \times n$ identity matrix. $p_{M(x)}(\lambda)$ is the so-called characteristic polynomial of matrix $M(x)$. From algebra, we know that $p_{M(x)}$ has n complex roots. Denote these roots with $\lambda_1(x), \cdots, \lambda_n(x) \in \mathbb{C}$, the polynomial can be represented by

$$p_{M(x)}(\lambda) = (\lambda_1(x) - \lambda) \cdots (\lambda_n(x) - \lambda). \tag{2.3}$$

$\lambda_1(x), \cdots, \lambda_n(x)$ are called the eigenvalues of matrix $M(x)$. Note that each eigenvalue $\lambda_i(x)$, $i = 1, \cdots, n$, is a function of x, if matrix $M(x)$ depends on x.

The spectrum of matrix $M(x)$ is defined as a finite set, which contains all n eigenvalues of $M(x)$. It can be denoted as

$$\Lambda_{M(x)}(x) := \{\lambda_i(x), i = 1, \cdots, n\}. \tag{2.4}$$

Note that, since matrix $M(x)$ is a function of x, its spectrum is also a function of x.

Eigenvalue functions can be generally denoted as

$$\phi(\lambda_1(x), \cdots, \lambda_n(x)), \tag{2.5}$$

where $\phi : \mathbb{C}^n \to \mathbb{R}$ denotes a smooth mapping. For example, an eigenvalue function of the spectral radius of matrix $M(x)$ can be formulated as

$$\rho(x) := \max_{i=1,\cdots,n} |\lambda_i(x)|. \tag{2.6}$$

In this work we consider the spectral abscissa (SA) function of matrix $M(x)$, defined by

$$\alpha_{M(x)}(x) := \max_{i=1,\cdots,n} Re(\lambda_i(x)). \tag{2.7}$$

$Re(\lambda_i)$ denotes the real part of λ_i, $i = 1, \cdots, n$. $\alpha : \mathbb{R}^m \to \mathbb{R}$ is a real-valued function. Note that, we sometimes write α_{M^*} to denote the spectral abscissa of a constant matrix $M^* \in \mathcal{M}_n$.

One of the most important properties of function $\alpha_{M(x)}(x)$ in Eq. (2.7) is that, it is continuous, but non-Lipschitz continuous [25]. Function $\alpha_{M(x)}(x)$ is non-smooth at certain points and the gradients at these points may approach infinity (i.e., the function may get infinitely steep). An exemplary non-Lipschitz continuous function is $\sqrt{x}$ in domain $[0, 1]$. Function $\sqrt{x}$ gets infinitely steep at $x = 0$. We present next an example to demonstrate the non-smoothness of the SA function. Related theoretical results on the smoothness of SA functions will be reviewed.

https://doi.org/10.51202/9783186952035 - Generiert durch IP 216.73.216.36, am 18.01.2026, 21:11:22. © Urheberrechtlich geschützter Inhalt. Ohne gesonderte Erlaubnis ist jede urheberrechtliche Nutzung untersagt, insbesondere die Nutzung des Inhalts im Zusammenhang mit, für oder in KI-Systemen, KI-Modellen oder Generativen Sprachmodellen.

Example 2.1. *We demonstrate non-Lipschitz continuity of the SA function of a non-symmetric matrix through a simple example adopted from [25]. Consider*

$$M_0(x) = \begin{pmatrix} 0 & 1 \\ -1 & -x \end{pmatrix},$$

where $x \in \mathbb{R}$.

The SA of $M_0(x)$ *is plotted in Fig. 2.1 as a function of* $x \in [1, 3]$. *We see that* $\alpha_{M_0(x)}(x)$ *is a smooth function of* x *almost everywhere, except for* $x = 2$. *At this non-smooth point* $x = 2$ *the function is non-Lipschitz continuous. If we examine the eigenvalues of* $M_0(x = 2)$, *we find that* $M_0(2)$ *has two repeated eigenvalues, i.e.,* $\lambda_1 = \lambda_2 = -1$.

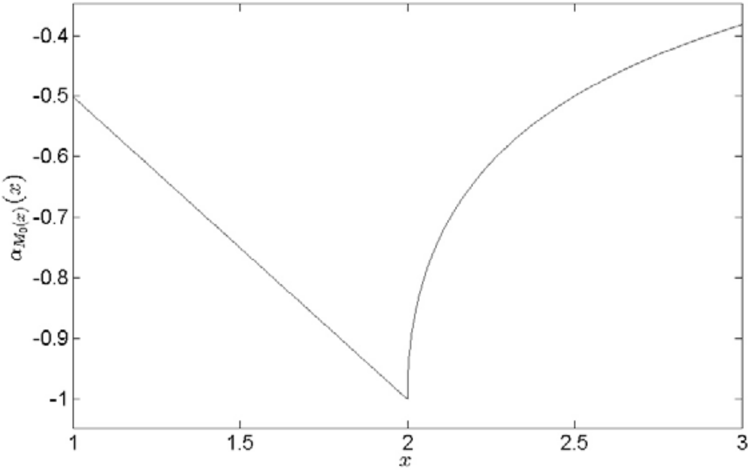


Figure 2.1: Spectral abscissa $\alpha_{M_0(x)}(x)$.

□

In the following lemma we establish continuity of the SA function $\alpha_{M(x)}(x)$.

Lemma 2.2.1 (Continuity of $\lambda_i(x)$). *If* $M : \mathbb{R}^m \to \mathcal{M}_n$ *is a continuous function, then all eigenvalue functions* $\lambda_i(x)$, $i = 1, \cdots, n$, *are locally continuous.*

Proof. Eigenvalues λ_i of matrix $M(x)$ are the roots of polynomial $p_{M(x)}(\lambda)$. Because M is a continuous function, all element functions $M_{i,j}(x)$, $i, j = 1, \cdots, n$ are continuous, all coefficients in the polynomial $p_{M(x)}(\lambda)$ are continuous. Hence, the roots of $p_{M(x)}(\lambda)$ are also continuous as shown in Theorem 1.4 in [112]. □

From the above lemma, because $\alpha_{M(x)}(x)$ is the largest of a finite number of continuous functions $Re(\lambda_i(\cdot))$, $i = 1, \cdots, n$, $\alpha_{M(x)}(x)$ is also continuous.

https://doi.org/10.51202/9783186952035 - Generiert durch IP 216.73.216.36, am 18.01.2026, 21:11:22. © Urheberrechtlich geschützter Inhalt. Ohne gesonderte Erlaubnis ist jede urheberrechtliche Nutzung untersagt, insbesondere die Nutzung des Inhalts im Zusammenhang mit, für oder in KI-Systemen, KI-Modellen oder Generativen Sprachmodellen.

To derive sufficient conditions for the smoothness of the SA function, we first introduce the definition of simple eigenvalues. We use x^* to denote a fixed point, while use i^* to denote a fixed index.

Definition 2.2.1 (Algebraic multiplicity)**.** *For* $i \in \{1, \cdots, n\}$*, the algebraic multiplicity of eigenvalue* $\lambda_i(x^*)$ *is defined as the multiplicity of root* $\lambda_i(x^*)$ *of polynomial* $p_{M(x^*)}(\lambda)$*, which is the largest integer* k^* *such that* $(\lambda_i(x^*) - \lambda)^{k^*}$ *appears on the right hand side of Eq. (2.3).*

Definition 2.2.2 (Simple eigenvalue)**.** *An eigenvalue* $\lambda_i(x^*)$*,* $i \in \{1, \cdots, n\}$*, is called a simple eigenvalue of matrix* $M(x^*)$*, if its algebraic multiplicity is one.*

Simple eigenvalues are sometimes called non-repeated eigenvalues (in the sense of isolated roots of the polynomial $p_{M(x^*)}(\lambda)$).

The following theorem gives a sufficient condition for the smoothness of the eigenvalue functions $\lambda_{i^*}(x)$, $i^* \in \{1, \cdots, n\}$. One may also refer to Theorem 2.1 in [168].

Theorem 2.2.2 (Smoothness of $\lambda_i(x)$, a specialization of Theorem 2.1 in [4])**.** *If* $\lambda_{i^*}(x^*)$*,* $i^* \in \{1, \cdots, n\}$*, is a simple eigenvalue of* $M(x^*)$*, and all elements in* $M(x)$ *are smooth functions of* x*, then* $\lambda_{i^*}(x)$ *is locally smooth near* x^**.*

However, simple smooth eigenvalues do not directly result in the smoothness of the SA function, because the spectrum $\Lambda_{M(x)}(x)$ in Eq. (2.4) may contain several simple eigenvalues, whose real parts are the same and equal to $\alpha_{M(x)}(x)$. For this reason, we need the definition of active eigenvalues to derive a sufficient condition for the smoothness of SA function.

Definition 2.2.3 (Active eigenvalues)**.** *An eigenvalue* $\lambda_{i^*}(x^*)$*,* $i^* \in \{1, \cdots, n\}$*, of matrix* $M(x^*) \in \mathcal{M}_n$ *is active, if* $Re(\lambda_{i^*}(x^*)) = \alpha_{M(x^*)}(x^*)$*.*

Definition 2.2.4 (Lexicographic order)**.** *For any pair of complex numbers* $a, b \in \mathbb{C}$*, we denote* $a \leq b$ *to represent a lexicographic order of* a *and* b*. It means that either*

$$\begin{aligned} &(i)\ Re(a) < Re(b), or \\ &(ii)\ Re(a) = Re(b), Im(a) \leq Im(b). \end{aligned}$$

$Re(\cdot)$ *and* $Im(\cdot)$ *denote the real part and the imaginary part of a complex number.*

Based on this definition, the sign "$\leq$" will be used to denote lexicographic order of complex numbers in the following.

Condition 2.2.1 (A sufficient condition for the smoothness of the SA function)**.** *Active eigenvalues of matrix* $M(x^*)$ *are either a real eigenvalue or a pair of conjugate complex eigenvalues. Hence, without loss of generality, if we assume that all eigenvalues are in lexicographic order, i.e.,*

$$\lambda_1(x^*) \leq \cdots \leq \lambda_n(x^*), \tag{2.8}$$

we have either

$$Re(\lambda_1(x^*)) \leq \cdots \leq Re(\lambda_{n-1}(x^*)) < Re(\lambda_n(x^*)),\ \ if\ \lambda_n(x^*) \in \mathbb{R},$$

or

$$Re(\lambda_1(x^*)) \leq \cdots Re(\lambda_{n-2}(x^*)) < Re(\lambda_{n-1}(x^*)) = Re(\lambda_n(x^*)),\ \ if\ \lambda_n(x^*) \in \mathbb{C}/\mathbb{R}.$$

https://doi.org/10.51202/9783186952035 - Generiert durch IP 216.73.216.36, am 18.01.2026, 21:11:22. © Urheberrechtlich geschützter Inhalt. Ohne gesonderte Erlaubnis ist jede urheberrechtliche Nutzung untersagt, insbesondere die Nutzung des Inhalts im Zusammenhang mit, für oder in KI-Systemen, KI-Modellen oder Generativen Sprachmodellen.

Corollary 2.2.3 (Smoothness of the SA function)**.** *If Condition 2.2.1 is satisfied at* $x = x^*$*, then the SA function* $\alpha_{M(x)}(x)$ *is locally smooth at* $x = x^*$*.*

Proof. Condition 2.2.1 implies:
(1) If $\lambda_n(x^*) \in \mathbb{R}$, then $\lambda_n(x^*)$ is a simple eigenvalue. In this case, from the continuity of eigenvalue functions (cf. refer to Lemma 2.2.1), there exists a neighborhood U of x^*, such that

$$\alpha_{M(x)}(x) = Re(\lambda_n(x)), \forall x \in U. \tag{2.9}$$

(2) If $\lambda_n(x^*) \in \mathbb{C}/\mathbb{R}$, $\lambda_n(x^*)$ and $\lambda_{n-1}(x^*)$ are two simple eigenvalues, or more precisely, a pair of conjugate complex numbers. Because of the continuity of each eigenvalue function, there exist a neighborhood U such that

$$max\{Re(\lambda_i(x)), i = 1, \cdots, n-2\} < Re(\lambda_{n-1}(x)) = Re(\lambda_n(x)), \forall x \in U.$$

Hence we have

$$\alpha_{M(x)}(x) = Re(\lambda_{n-1}(x)) = Re(\lambda_n(x)), \forall x \in U. \tag{2.10}$$

Smoothness of $\alpha_{M(x)}(x)$ follows from Eqs. (2.9), (2.10) by using the smoothness property of simple eigenvalues established in Theorem 2.2.2. □

In summary, we have shown that although the SA function $\alpha_{M(x)}(x)$ is in general non-Lipschitz continuous (cf. Example 2.1), it is locally smooth under Condition 2.2.1.

Although the SA function is generally non-smooth at certain points, smoothness can be expected typically at almost all points of the domain. We discuss next the expressions to calculate the gradients of the SA function at smooth points.

Under the conditions of Corollary 2.2.3, first- and higher-order derivatives of the SA function $\alpha_{M(x)}(x)$ can be derived straightforwardly from the sensitivity analysis of simple eigenvalue functions $\lambda_i(x)$, $i = 1, \cdots, n$, which have been discussed in [123]. Here, we present a method based on the left and right eigenvectors. Note that the characterization of the variational properties of SA functions at non-smooth points is still an active field of research and therefore out of the scope of this work (cf. Section 5.4.2).

For $x = x^*$ and $i^* \in \{1, \cdots, n\}$, let $v_{i^*}, u_{i^*} \in \mathbb{C}^n$ be the right and left column eigenvectors of matrix $M(x^*)$ with respect to eigenvalue $\lambda_{i^*}(x^*)$. Hence, v_{i^*} and u_{i^*} satisfy

$$\begin{aligned} u_{i^*}^T M(x^*) &= \lambda_{i^*} u_{i^*}^T, \\ M(x^*) v_{i^*} &= \lambda_{i^*} v_{i^*}. \end{aligned}$$

Under the assumptions of Theorem 2.2.2, the first-order derivatives of $\lambda_{i^*}(x)$ at $x = x^*$ can be evaluated [123] to

$$\frac{\partial \lambda_{i^*}(x^*)}{\partial x_j} = \frac{u_{i^*}^T \frac{\partial M(x)}{\partial x_j}|_{x=x^*} v_{i^*}}{u_{i^*}^T v_{i^*}}, \; j = 1, \cdots, m. \tag{2.11}$$

x_j denotes the j-th element in vector x. $\frac{\partial M(x)}{\partial x_j} \in \mathbb{R}^{n \times n}$ is the element-by-element partial derivative of matrix $M(x)$ with respect to x_j. Note that, the derivatives are complex functions since $\lambda_{i^*}(x)$ is complex.

https://doi.org/10.51202/9783186952035 - Generiert durch IP 216.73.216.36, am 18.01.2026, 21:11:22. © Urheberrechtlich geschützter Inhalt. Ohne gesonderte Erlaubnis ist jede urheberrechtliche Nutzung untersagt, insbesondere die Nutzung des Inhalts im Zusammenhang mit, für oder in KI-Systemen, KI-Modellen oder Generativen Sprachmodellen.

If the conditions in Corollary 2.2.3 are fulfilled, the first-order derivatives of the SA function $\alpha_{M(x)}(x)$ at $x = x^*$ can be evaluated from Eq. (2.11) to

$$\frac{\partial \alpha_{M(x)}(x^*)}{\partial x_j} = Re\frac{\partial \lambda_n(x^*)}{\partial x_j} = Re(\frac{u_n^T \frac{\partial M(x)}{\partial x_j}|_{x=x^*} v_n}{u_n^T v_n}),\ j = 1, \cdots, m. \tag{2.12}$$

As it has been defined in Eq. (2.11), $v_n, u_n \in \mathbb{C}^n$ denote the right and left column eigenvectors of matrix $M(x^*)$ with respect to eigenvalue $\lambda_n(x^*)$, which is ordered lexicographically by Eq. (2.8).

2.3 Lyapunov stability

Lyaponov stability is a concept related to the equilibrium points of system (2.1). A point $x^* \in \mathbb{R}^m$ is called an equilibrium point of system (2.1), if $0 = f(x^*)$ holds. Without loss of generality, i.e. after transforming the coordinates, we can always assume $x^* = 0$. Equilibrium points are also called steady states of a dynamic system.

An equilibrium point is stable, if all solutions starting from nearby initial points stay in a neighborhood of this equilibrium point. An equilibrium point is asymptotically stable, if all solutions starting from nearby points stay not only in a neighborhood, but also converge to the equilibrium point as time goes to infinity. This concept leads to the following definitions:

Definition 2.3.1 (Stability). *An equilibrium point $x^* = 0$ is stable, if for any $\epsilon > 0$, there exists a positive number $\delta = \delta(\epsilon)$, such that*

$$\|x(0)\| < \delta \ \Rightarrow\ \|x(t)\| < \epsilon, \forall t \geq 0.$$

An equilibrium point is called unstable, if it is not stable.

Definition 2.3.2 (Asymptotical stability). *An equilibrium point $x^* = 0$ is asymptotically stable, if there exists a positive number δ, such that*

$$\|x(0)\| < \delta \ \Rightarrow\ \|x(t)\| \to x^*,\ as\ t \to \infty.$$

Checking the stability of an equilibrium point using these definitions requires a solution of system (2.1) for infinitely large t, which is inconvenient in practice. An attractive stability result is based on the following theorem.

Theorem 2.3.1 (Lyaponov stability, Theorem 3.1 in [84]). *Let $x^* = 0$ be an equilibrium point for system (2.1) and $D \subset \mathbb{R}^m$ be a domain containing x^*. Let $V : D \to \mathbb{R}$ a continuously differentiable function, such that*

$$V(0) = 0\ and\ V(x) > 0\ in\ D/\{0\},$$
$$\dot{V}(x) \leq 0\ in\ D.$$

Then, x^* is stable. Moreover, if

$$\dot{V}(x) < 0\ in\ D,$$

then x^* is asymptotically stable.

https://doi.org/10.51202/9783186952035 - Generiert durch IP 216.73.216.36, am 18.01.2026, 21:11:22. © Urheberrechtlich geschützter Inhalt. Ohne gesonderte Erlaubnis ist jede urheberrechtliche Nutzung untersagt, insbesondere die Nutzung des Inhalts im Zusammenhang mit, für oder in KI-Systemen, KI-Modellen oder Generativen Sprachmodellen.

Note that the conditions in the above theorem are sufficient conditions. The theorem provides an elegant way to check (asymptotic) stability. However, there does not exist a systematic method to find Lyapunov functions for general dynamic systems [84].

Stability and asymptotic stability relate to the eigenvalues of the Jacobian matrix $\partial f/\partial x$ of system (2.1). To see this relationship, let use first consider the linear case in the form of a linear ODE

$$\dot{x} = Ax,\ x(0) = x_0. \tag{2.13}$$

$A \in \mathbb{R}^{m\times m}$ is a given constant matrix. A is the Jacobian matrix of system (2.13).

Denote $\lambda_1, \cdots, \lambda_m \in \mathbb{C}$ as m eigenvalues of A. For any matrix A, there is a non-singular matrix $P \in \mathbb{C}^{m\times m}$, so that

$$P^{-1}AP = \text{block diag}(J_1, \cdots, J_p).$$

J_1, $\cdots$, J_p are the so-called Jordan blocks, which are in the form of

$$J_j = \begin{bmatrix} \lambda_j & 1 & & \\ & \lambda_j & \ddots & \\ & & \ddots & 1 \\ & & & \lambda_j \end{bmatrix} \in \mathbb{C}^{m_j\times m_j},\ \forall j = 1, \cdots, p$$

From the following theorem, stability and asymptotic stability of system (2.13) can be assumed by inspection of the eigenvalues of A.

Theorem 2.3.2 (Lyaponov stability for linear systems, Theorem 3.5 in [84]). *The equilibrium point $x^* = 0$ of system (2.13) is stable, iff all eigenvalues of A satisfy $Re(\lambda_i) \le 0$, $i \in \{1, \cdots, m\}$ and every eigenvalue with $Re(\lambda_i) = 0$, $i \in \{1, \cdots, m\}$, has an associated Jordan block of dimension one-by-one. The equilibrium point $x^* = 0$ is (globally) asymptotically stable, iff all eigenvalues of A satisfy $Re(\lambda_i) < 0$, $i = 1, \cdots, m$.*

If all eigenvalues of A satisfy $Re(\lambda_i) < 0$, matrix A is called a stability or a Hurwitz matrix.

Now let us extend Theorem 2.3.2 to nonlinear dynamic systems (2.1). The link can be built by using the following theorem, which is not restricted to stability analysis. The theorem says that, near a (not necessarily stable) hyperbolic equilibrium point[1] x^*, the solutions of the nonlinear system (2.1) have the same qualitative properties as the ones of the linear system (2.13) with

$$A = \frac{\partial f}{\partial x}|_{x=x^*}. \tag{2.14}$$

Hence, eigenvalue-based conditions in Theorem 2.3.2 for linear systems can be extended to check local (asymptotical) stability of the original nonlinear system.

To formally introduce the theorem, we assume that for each initial point $x_0 \in D'$, system (2.1) has a unique global solution $x(t) = \phi_t(x_0)$ for all $t > 0$ (refer to Theorem 2.1.5 for the existence of global solutions). We use the function $\phi_t(x_0)$ to explicitly denote the dependence of solution $x(t)$ on the selected initial point x_0. That is, $\phi_t(x_0)$ denotes the solution/trajectory of system (2.1), which is started from $x(0) = x_0$. $\phi_t(x_0)$ is also called a flow of system (2.1). Furthermore, a function $\Psi : X \to Y$ between two topological spaces X and Y is called a homeomorphism, if Ψ is a continuous bijection (one-to-one and onto) and its inverse function Ψ^{-1} is also continuous. If a homeomorphism exists between two topological spaces, these two spaces are said to be homeomorphic.

---

[1] A equilibrium point is called hyperbolic, if the Jacobian matrix of system (2.1) has no eigenvalues with zero real parts at this point.

https://doi.org/10.51202/9783186952035 - Generiert durch IP 216.73.216.36, am 18.01.2026, 21:11:22. © Urheberrechtlich geschützter Inhalt. Ohne gesonderte Erlaubnis ist jede urheberrechtliche Nutzung untersagt, insbesondere die Nutzung des Inhalts im Zusammenhang mit, für oder in KI-Systemen, KI-Modellen oder Generativen Sprachmodellen.

Theorem 2.3.3 (Hartman-Grobman Theorem, refer to Section 2.8 in [132])**.** *Assume that the origin* $x^* = 0$ *is an equilibrium of system (2.1) and* $f(\cdot)$ *is continuous differentiable. Suppose that* A *in Eq. (2.14) has no eigenvalues with zero real parts, then there exists a homeomorphism* Ψ *of an open set* $U \subset \mathbb{R}^m$ *onto an open set* $V \subset \mathbb{R}^m$ *which contains the origin, such that for each* $x_0 \in U$ *there is an open interval* $I_0 = [0, t_1]$, $t_1 > 0$, *such that for* $\forall x_0 \in U$ *and* $\forall t \in I_0$

$$\Psi \circ \phi_t(x_0) = e^{At}\Psi(x_0).$$

This theorem says that Ψ maps the trajectories of system (2.1) near the origin onto the trajectories of system (2.13) near the origin, and it also preserves the parametrization in time. Hence, if Theorem 2.3.3 holds, stability of the nonlinear system (2.1) can be assumed by checking the stability of the linear system (2.13) with A defined in Eq. (2.14).

Corollary 2.3.4 (Lyapunov's indirect method, refer to Theorem 3.7 in [84])**.** *Consider* $x^* = 0$ *is an equilibrium point of system (2.1), let* A *be defined in Eq. (2.14). Then the following statements hold:*

1. *If all eigenvalues of* A *have negative real parts, then the origin of system (2.1) is asymptotically stable.*
2. *If one or more of the eigenvalues of* A *has positive real parts, then the origin of system (2.1) is unstable.*

According to this corollary, stability of nonlinear systems can be concluded by computing the eigenvalues of matrix A. Note, if any eigenvalue has zero real part, stability can not be decided using this corollary. In this case, more information, e.g. the Hessian matrix of f, about the system is needed.

Corollary 2.3.4 is the basic theoretical foundation of this work, because one of our design goals is to guarantee asymptotic stability of a designed equilibrium point of a reactor network. In particular, we are looking for a reactor network and an equilibrium point x^*, which minimizes a cost function, such that all eigenvalues of matrix A in Eq. (2.14) have negative real parts.

2.4 Dynamic response

Consider nonlinear system (2.1), denote

$$A(x) = \frac{\partial f}{\partial x}(x) \tag{2.15}$$

as the Jacobian matrix of system (2.1), which depends on the evaluation point $x \in \mathbb{R}^m$. According to Corollary 2.3.4 and the notations introduced in Section 2.2, system (2.1) is asymptotically stable at an equilibrium point x^*, if the constraint on the SA,

$$\alpha_{A(x)}(x) = \max_{i=1,\cdots,m} Re(\lambda_i(x)) < 0 \tag{2.16}$$

holds for $x = x^*$. $\lambda_i(x)$, $i = 1, \cdots, m$, denotes the eigenvalues of Jacobian matrix $A(x) \in \mathbb{R}^{m \times m}$. In this section, we discuss the dynamic properties of system (2.1), which is ensured by imposing a negative upper bound $-c < 0$ to $\alpha_{A(x)}(x)$, i.e.

$$\alpha_{A(x)}(x) = \max_{i=1,\cdots,m} Re(\lambda_i)(x) \leq -c, \tag{2.17}$$

https://doi.org/10.51202/9783186952035 - Generiert durch IP 216.73.216.36, am 18.01.2026, 21:11:22. © Urheberrechtlich geschützter Inhalt. Ohne gesonderte
Erlaubnis ist jede urheberrechtliche Nutzung untersagt, insbesondere die Nutzung des Inhalts im Zusammenhang mit, für oder in KI-Systemen, KI-Modellen oder Generativen Sprachmodellen.

where $c > 0$ is a selected real constant.

The following theorem states that c determines the response speed of system (2.1), initialized from a nearby point.

Theorem 2.4.1 (Refer to Section 2.9 in [132]). *If Eq. (2.17) holds for a steady-state* x^* *of system (2.1), for any given* $\epsilon > 0$, *there exist* $\delta > 0$ *such that if* $\|x_0 - x^*\| \leq \delta$,

$$\|\phi_t(x_0) - x^*\| < \epsilon\, e^{-ct}, \forall t > 0.$$

Hence, one can use c to measure how fast a trajectory converges to x^*, if system (2.1) is initialized from x_0 near x^*.

The consequence of Eq. (2.17) can be also illustrated from a different perspective. Consider that $f(\cdot)$ in Eq. (2.1) also depends on an input vector $u \in \mathbb{R}^{n_u}$:

$$\dot{x} = f(x, u),\ x(0) = x_0. \tag{2.18}$$

After linearizing this system at a given steady state $(x^{*T}, u^{*T})^T$, the linearized system can be denoted as

$$\Delta\dot{x} = A\Delta x + B\Delta u(t),\ \ \Delta x(0) = x_0', \tag{2.19}$$

where $\Delta x(t) = x(t) - x^*$, $\Delta u(t) = u(t) - u^*$ and $x_0' = x_0 - x^*$. $A = \partial f/\partial x(x^*, u^*) \in \mathbb{R}^{m \times n_x}$ is the Jacobian matrix evaluated at $(x^{*T}, u^{*T})^T$, while $B = \partial f/\partial u(x^*, u^*) \in \mathbb{R}^{m \times n_u}$ is the input gain matrix evaluated at $(x^{*T}, u^{*T})^T$. To simply the notation, we can shift the origin of the coordinates to $(x^{*T}, u^{*T})^T$, so that the symbol Δ in Eq. (2.19) can be omitted. This results in an equivalent linear system

$$\dot{x} = Ax + Bu(t),\ \ x(0) = x_0'. \tag{2.20}$$

Note that x, u in Eq. (2.20) are not the same ones in Eq. (2.18).

The analytical solution of Eq. (2.20) is [106]

$$x(t) = e^{At}x_0' + \int_0^t e^{A(t-\tau)}Bu(\tau)d\tau. \tag{2.21}$$

To illustrate the ensured dynamic response speed by using Eq. (2.17), we consider the solution of the system (2.20) under impulsive and step inputs. If impulsive inputs

$$u(t) = u_d(t) = (\delta(t), \cdots, \delta(t))^T \in \mathbb{R}^{n_u} \tag{2.22}$$

are applied, where $\delta(t)$ denotes the Dirac delta function, we get from Eq. (2.21)

$$\begin{aligned} x(t) =& e^{At}x_0' + \int_0^t e^{A\tau}Bu_d(\tau - t)d\tau \\ =& e^{At}x_0' + e^{At}\underbrace{B\,\mathbf{1}}_{:=b_0} \\ =& e^{At}(x_0' + b_0), \end{aligned} \tag{2.23}$$

https://doi.org/10.51202/9783186952035 - Generiert durch IP 216.73.216.36, am 18.01.2026, 21:11:22. © Urheberrechtlich geschützter Inhalt. Ohne gesonderte Erlaubnis ist jede urheberrechtliche Nutzung untersagt, insbesondere die Nutzung des Inhalts im Zusammenhang mit, für oder in KI-Systemen, KI-Modellen oder Generativen Sprachmodellen.

where $\mathbf{1} \in \mathbb{R}^{n_u \times 1}$ is the n_u-dimensional column vector containing only $1 \in \mathbb{R}$ at every position. Note that the property

$$\int_{-\infty}^{\infty} g(\tau)\delta(\tau - t)d\tau = g(t)$$

has been used for the smooth function $g(\cdot) : \mathbb{R} \rightarrow \mathbb{R}$.

If step inputs are applied, i.e.,

$$u(t) = u_s(t) = \begin{cases} (1, \cdots, 1)^T \in \mathbb{R}^{n_u}, \text{if } t \geq 0 \\ (0, \cdots, 0)^T, \text{otherwise} \end{cases} \tag{2.24}$$

denotes a vector of unit step functions. The solution of system (2.20) becomes

$$\begin{aligned} x(t) =& e^{At}x_0' + \int_0^t e^{A(t-\tau)}b_0 d\tau \\ =& e^{At}x_0' + A^{-1}(e^{At}b_0 - b_0). \end{aligned} \tag{2.25}$$

Lemma 2.4.2 (Corollary in Section 1.8 of [132]). *$\forall c_0 \in \mathbb{R}^m$, $\forall A \in \mathbb{R}^{m \times m}$, each element in vector $e^{At}c_0$ is a linear combination of*

$$\eta_{i,k}(t) = t^k e^{Re(\lambda_i)t} cos(Im(\lambda_i)t),\ \forall k = 0, \cdots, m-1, \forall i = 1, \cdots, m, \tag{2.26}$$

and

$$\eta'_{i,k}(t) = t^k e^{Re(\lambda_i)t} sin(Im(\lambda_i)t),\ \forall k = 0, \cdots, m-1, \forall i = 1, \cdots, m. \tag{2.27}$$

λ_i, $i = 1, \cdots, m$, denote the eigenvalues of matrix A. $Re(\lambda_i)$ and $Im(\lambda_i)$ denote the real and imaginary parts of λ_i, respectively.

Consequently, solutions of system (2.20) can be written as

$$x(t) - x_s^* = \Sigma \underbrace{(\cdots, \eta_{i,k}(t), \cdots, \eta'_{i,k}(t), \cdots)^T}_{:=\eta(t)},$$

if impulsive and step inputs are applied. $x_s^* \in \mathbb{R}^m$ refers to the new steady state after applying $u_d(t)$ or $u_s(t)$:

$$x_s^* = \begin{cases} 0, & \text{if impulse inputs } u_d(t) \text{ are applied,} \\ -A^{-1}b_0, & \text{if step inputs } u_s(t) \text{ are applied.} \end{cases}$$

$\Sigma \in \mathbb{R}^{m \times 2m^2}$ denotes a constant matrix, which is determined by the right hand sides of Eqs. (2.23) or (2.25), respectively.

Consider that Eq. (2.17) is fulfilled for a certain $c > 0$, we then have

$$\|x(t) - x_s^*\|_\infty = \|\Sigma\ \eta(t)\|_\infty \leq \|\Sigma\|_\infty\ \|\eta(t)\|_\infty \leq \|\Sigma\|_\infty \frac{\max\{t^0, \cdots, t^{m-1}\}}{e^{ct}},\ \forall t \geq 0.$$

https://doi.org/10.51202/9783186952035 - Generiert durch IP 216.73.216.36, am 18.01.2026, 21:11:22. © Urheberrechtlich geschützter Inhalt. Ohne gesonderte Erlaubnis ist jede urheberrechtliche Nutzung untersagt, insbesondere die Nutzung des Inhalts im Zusammenhang mit, für oder in KI-Systemen, KI-Modellen oder Generativen Sprachmodellen.

Table 2.1: Estimating the decay time t_d from the spectral abscissa $-c$ by using Eq. (2.31) for $\mu = 0.01$.

	$c = 10^{-4}$	$c = 10^{-3}$	$c = 10^{-2}$	$c = 10^{-1}$
t_d [s]	46051.70	4605.17	460.51	46.05

It means that, in both cases the convergence to the new steady states x_s^* is bounded by term $p_0(t)/e^{ct}$, $p_0(t) := \|\Sigma\|_\infty \max\{t^0, \cdots, t^{m-1}\}$. When c is close to zero, $x(t)$ decays slowly to the new steady state x_s^*. When $c > 0$ is far away from zero, the system converges to the new steady state quickly.

In order to estimate the decay time for a given c, refer also to Section 3.4.3 in [49], we consider the scalar linear system

$$\dot{\hat{x}} = -c\hat{x} + b\hat{u}, \hat{x}(0) = 0, \tag{2.28}$$

where $\hat{x}(t)$, $\hat{u}(t)$, $b \in \mathbb{R}$. For this system, $-c$ denotes both the Jacobian matrix and its spectral abscissa. After given a non-unit step input, i.e., $\hat{u}(t) = 0$, for $t < 0$, and $\hat{u}(t) = \bar{u} > 0$, for $t \geq 0$, the solution of Eq. (2.28) is

$$\hat{x}(t) - \hat{x}^* = -\hat{x}^* e^{-ct}, \tag{2.29}$$

where $\hat{x}^* = b\bar{u}/c$ denotes the new steady state. For $\mu \in (0, 1]$ as a given quantity, we define the decay time t_d such that the error $|\hat{x}(t_d) - \hat{x}^*|$ is equal to a fraction μ of the absolute value of $\hat{x}^*$, i.e.,

$$|\hat{x}(t_d) - \hat{x}^*| = \mu \, |\hat{x}^*|. \tag{2.30}$$

Using Eqs. (2.29), (2.30),

$$t_d = \frac{-ln \, \mu}{c} \tag{2.31}$$

relates the decay time t_d to the spectral abscissa $-c$ of system (2.28).

Eq. (2.31) can be used pragmatically to estimate the decay time t_d from the spectral abscissa $-c$ of Jacobian matrix A also for higher-dimensional systems. Table 2.1 lists the the computed decay time t_d for different value of c.

To summarize, we have illustrated that eigenvalue constraint (2.17) can be used to guarantee specified response speed of nonlinear system (2.1). The larger c in Eq. (2.17), the shorter the decay time and therefore the faster the response. Note that, a positive c guarantees automatically asymptomatic stability of the given steady state.

2.5 Extension to differential-algebraic systems

Before we extend the results for ODE to differential-algebraic equations (DAE), we first introduce the Implicit Function Theorem. This theorem is one of the most fundamental results in applied mathematics.

https://doi.org/10.51202/9783186952035 - Generiert durch IP 216.73.216.36, am 18.01.2026, 21:11:22. © Urheberrechtlich geschützter Inhalt. Ohne gesonderte
Erlaubnis ist jede urheberrechtliche Nutzung untersagt, insbesondere die Nutzung des Inhalts im Zusammenhang mit, für oder in KI-Systemen, KI-Modellen oder Generativen Sprachmodellen.

Theorem 2.5.1 (Implicit Function Theorem, in Section 2.1.2 of [84])**.** *Assume that* $f : \mathbb{R}^m \times \mathbb{R}^n \to \mathbb{R}^m \in \mathcal{C}^k$, $k \geq 1$, *i.e.,* f *is* k*-th order continuously differentiable, at each point* (x, y) *of an open set* $D \subset \mathbb{R}^m \times \mathbb{R}^n$*. Let* (x^*, y^*) *be a point in* D *for which* $f(x^*, y^*) = 0$ *and for which the Jacobian matrix* $\frac{\partial f}{\partial x}(x^*, y^*)$ *is non-singular. Then, there exist neighborhoods* U *of* x^* *and* V *of* y^* *such that for each* $y \in V$ *the equation* $f(x, y) = 0$ *has a unique solution for* $x \in U$*. Moreover, this solution can be denoted by* $x = g(y)$*, i.e.,*

$$f(g(y), y) \equiv 0, \forall y \in V,$$

where $g \in \mathcal{C}^k$ *at* $y = y^*$*, i.e., function* $g(\cdot)$ *is also* k*-th order continuously differentiable.*

The Implicit Function Theorem says that, if the Jacobian matrix is non-singular, an equation system $0 = f(x, y)$ locally determines a function $x = g(y)$ and the smoothness of this function $g(\cdot)$ is the same as the smoothness of $f(\cdot)$.

Now we apply the Implicit Function Theorem to extend the results of ODE to a special class of DAE systems, given as

$$\begin{aligned} \dot{x}^d &= f^d(x^d, x^a),\ x^d(0) = x^d_0, \\ 0 &= f^a(x^d, x^a), \end{aligned} \tag{2.32}$$

where $x^d \in \mathbb{R}^m$ and $x^a \in \mathbb{R}^n$ are the so-called differential and algebraic states. The functions $f^d : \mathbb{R}^m \times \mathbb{R}^n \to \mathbb{R}^m$ and $f^a : \mathbb{R}^m \times \mathbb{R}^n \to \mathbb{R}^n$, referring to the differential and algebraic equations, are assumed to be sufficiently smooth. System (2.32) is called a semi-explicit DAE system.

Although DAE systems are in general different from ODE systems [134], under certain conditions semi-explicit DAE systems can be transformed equivalently to an ODE system. Therefore one can directly extend the results of ODE systems to this type of DAE systems. To do this, we assume that $x^{d*} \in \mathbb{R}^m$ and $x^{a*} \in \mathbb{R}^n$ satisfy $0 = f^a(x^{d*}, x^{a*})$ and

$$det(\frac{\partial f^a}{\partial x^a}|_{x^{d*}, x^{a*}}) \neq 0. \tag{2.33}$$

From the Implicit Function Theorem 2.5.1, the algebraic equation $0 = f^a(x^d, x^a)$ locally determines a sufficiently smooth function $g : V_{x^{d*}} \subset \mathbb{R}^m \to U_{x^{a*}} \subset \mathbb{R}^n$ such that

$$0 \equiv f^a(x^d, g(x^d)),\ \forall x^d \in V_{x^{d*}}.$$

Therefore, the solutions of differential states $x^d(t)$ can be represented locally by

$$\dot{x}^d = f^d(x^d, g(x^d)),\ x^d(0) = x^d_0. \tag{2.34}$$

Eq. (2.34) says that under condition (2.33) solutions of DAE system (2.32) are locally the same as the solutions of the ODE system (2.34). Moreover, this result can be extended globally, i.e., removing the condition about neighborhoods, by assuming that condition (2.33) holds for all $x^d \in \mathbb{R}^m$ and all $x^a \in \mathbb{R}^n$ on the solution trajectory. Hence, the existing theoretical results for ODE systems can be applied directly for semi-explicit DAE systems. In this work, we consider only DAE systems, which can be transformed to ODE systems by the above-mentioned procedures.

https://doi.org/10.51202/9783186952035 - Generiert durch IP 216.73.216.36, am 18.01.2026, 21:11:22. © Urheberrechtlich geschützter Inhalt. Ohne gesonderte Erlaubnis ist jede urheberrechtliche Nutzung untersagt, insbesondere die Nutzung des Inhalts im Zusammenhang mit, für oder in KI-Systemen, KI-Modellen oder Generativen Sprachmodellen.

3 Open-loop reactor network synthesis

Having presented an introduction and having reviewed related theoretical fundamentals of dynamic systems, we will start to present the major contents of this work. The discussion is organized in two parts. The first part (Chapter 3) is for reactor network design problems in open-loops and the second part (Chapter 4) is for simultaneous closed-loop design of reactor networks. Each chapter follows a similar way of presentation: first modeling and afterwards problem formulation. Solution strategies to solve the derived design problems will be discussed together in Chapter 5.

An example is introduced first, which will be used for illustration throughout the following chapter and will be solved as a case study in Chapter 6.

Example 3.1. *Allyl chloride can be produced by means of non-catalytic chlorination of propylene in the vapor phase [129]. The reaction mechanism is as follows:*

$$\begin{aligned}
&\underbrace{Cl_2}_{C} + \underbrace{CH_2 - CHCH_3}_{A} \overset{k_1}{\to} \underbrace{CH_2 - CHCH_2Cl}_{B} + HCl,\\
&\underbrace{Cl_2}_{C} + \underbrace{CH_2 - CHCH_2Cl}_{B} \overset{k_2}{\to} \; ClCH - CHCH_2Cl + HCl,\\
&\underbrace{Cl_2}_{C} + \underbrace{CH_2 - CHCH_3}_{A} \overset{k_3}{\to} \; CH_2ClCHClCH_3.
\end{aligned}$$

We use A, B, C to denote propylene, allyl chloride and chlorine. A, C are raw materials, B is the main product, while 1,3-dichloropropene and 1,2-dichloropropane are side products. The reaction kinetics is modeled as

$$\begin{aligned}
&r_1(c_A, T) = k_1 c_A c_C, \quad r_2(c_B, T) = k_2 c_B c_C, \quad r_3(c_A, T) = k_3 c_A c_C,\\
&k_1 = a_1 e^{-15840/RT}, \quad k_2 = a_2 e^{-23760/RT}, \quad k_3 = a_3 e^{-7920/RT}.
\end{aligned} \tag{3.1}$$

T $[K]$ denotes temperature, while c_A, c_B, c_C $[mol/m^3]$ denote concentrations of A, B, C, respectively. All parameters can be found in Table 3.1. The reaction rates for component A, B, C are

$$R_A = -r_1 - r_3, \; R_B = r_1 - r_2, \; R_C = -r_1 - r_2 - r_3.$$

□

3.1 Structured modeling of reactor networks

In this section, we present a structured dynamic model of open-loop reactor networks. The model allows an efficient treatment of eigenvalue-based dynamic properties for the design of open-loop reactor networks. A structured modeling approach will be presented below, which firstly models each individual reactor in a reactor network superstructure as a separate subsystem and then builds the flow connections to form a connected network.

https://doi.org/10.51202/9783186952035 - Generiert durch IP 216.73.216.36, am 18.01.2026, 21:11:22. © Urheberrechtlich geschützter Inhalt. Ohne gesonderte Erlaubnis ist jede urheberrechtliche Nutzung untersagt, insbesondere die Nutzung des Inhalts im Zusammenhang mit, für oder in KI-Systemen, KI-Modellen oder Generativen Sprachmodellen.

Table 3.1: Constants and reaction parameters for the allyl chloride example [129].

parameter	value		units
a_1	1.5×10^6	reaction constant	1/s
a_2	4.4×10^8	reaction constant	1/s
a_3	1.0×10^2	reaction constant	l/mol s
R	1.987	gas constant	cal/mol K
H_1	118.82	reaction heat	kJ/mol
H_2	114.79	reaction heat	kJ/mol
H_3	183.03	reaction heat	kJ/mol

3.1.1 A structured representation of reactor network models

Fig. 3.1 shows an open-loop superstructure of a N-reactor network [87], which will be used throughout this work. The maximal number of reactors N in the superstructure has to be fixed to a predefined number.

Raw materials are fed into the network and split into N reactor's feed flows. Each reactor, which can be either a CSTR or a PFR, has N inlet and N outlet flows. One of the inlets is used as the raw material feed and the other inlets are connected to the outlets of other reactors. All N reactor outlets have the same concentrations and temperature but different flowrates. One of them contributes to the network's outlet and the others feed other reactors. The product mixer on the right hand side of the network generates a product stream by mixing individual outlet streams of each of the reactors.

Except for the system's inlet and outlet, all other (internal) inlet and outlet streams are allowed to be removed from the superstructure. Likewise, not all reactors need to be used in the final design. When a reactor is not used in the designed network, it is called *idle*. All inlet and outlet streams of an idle reactor must show zero flowrates such that no material is moved into or outside of the reactor.

By deciding on the existence of inlet and outlet streams as well as of reactors, the superstructure realizes a rich set of structural alternatives with different kinds of bypass and recycle streams. A bypass can be realized by feeding a reactor's outlet to the product mixer. In this way, the outlet to the product mixer becomes a bypass for the other reactors. Recycles can be realized by first connecting several reactors in a sequence and then feeding the reacted material from the last reactor into the first reactor of the sequence. Some trivial structural alternatives such as reactors in series or in parallel are also contained in the superstructure.

The reactor network in Fig. 3.1 is interpreted as a system of $N+2$ subsystems shown in Fig. 3.2. We use $i = 1, \cdots, N$ to index each subsystem corresponding to each reactor (together with the mixer and the splitter before and after each reactor) in Fig. 3.1. Subsystem $N+1$ refers to the raw material splitter, while subsystem $N+2$ refers to the product mixer. For every subsystem, small gray boxes on the left represent "inlet ports", which are indexed by $k = 1, \cdots, N$. Likewise, small white boxes on the right of each subsystem represent "outlet ports" indexed by $j = 1, \cdots, N$. These ports abstract nozzles connecting the pipes to an apparatus. For subsystem i, $i = 1, \cdots, N$, each subsystem has N inlet ports and N outlet ports. Subsystem $N+1$ has an inlet port representing the inlet of the network and N outlet ports, while subsystem $N+2$ has N inlet ports and an outlet port representing the product outlet. Each subsystem is connected to others through solid

https://doi.org/10.51202/9783186952035 - Generiert durch IP 216.73.216.36, am 18.01.2026, 21:11:22. © Urheberrechtlich geschützter Inhalt. Ohne gesonderte
Erlaubnis ist jede urheberrechtliche Nutzung untersagt, insbesondere die Nutzung des Inhalts im Zusammenhang mit, für oder in KI-Systemen, KI-Modellen oder Generativen Sprachmodellen.

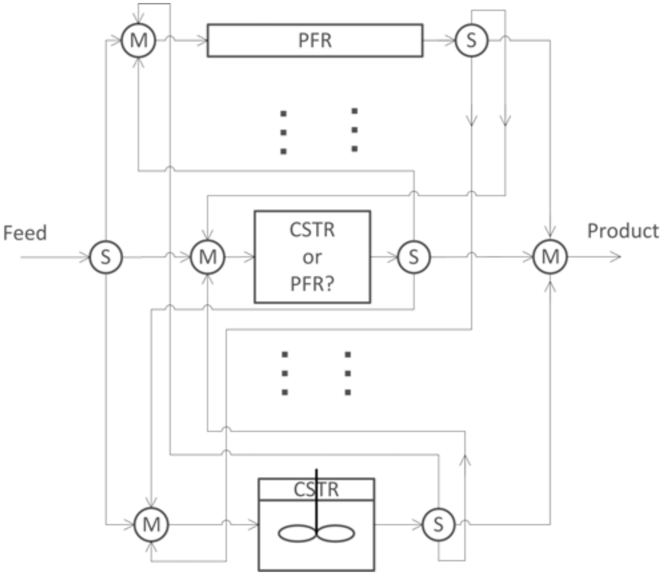


Figure 3.1: An open-loop superstructure of an N-reactor network [87]. M and S refer to mixing and splitting units. PFR and CSTR refer to plug flow reactor and continuous stirred-tank reactor, respectively.

arrows, which abstract pipes delivering material in the reactor network. A pipe always links an output port to an input port.

We define symbols to index ports and connections (pipes) as follows: (i, j) is used to indicate the j-th outlet port of subsystem i, while (i, k) is used to indicate the k-th inlet port of subsystem i. Here, symbol j and k always refer to inlet and outlet ports, respectively. For every outlet port (i, j), we use the mapping $l(i, j) = (i', k')$ to indicate the index of its connected inlet port (i', k'). (i', k') can be found by following a solid arrow starting from outlet port (i, j). Similarly, we use $h(i', k') = (i, j)$ to access the outlet port (i, j), which is connected to the inlet port (i', k'). The value of $h(i', k')$ can be found by backtracking a solid arrow pointing at inlet port (i', k'). We also use the term “connection” or “pipe” to refer to a material stream from an outlet port to an inlet port. A connection (pipe) from outlet port (i, j) to inlet port (i', k') is denoted as $(i, j) \triangleright (i', k')$. “$\triangleright$” here has the meaning of “to”:

$$\underbrace{(i,j)}_{\substack{\text{outlet}\\\text{port}}}\underbrace{\triangleright}_{\text{to}}\underbrace{(i',k')}_{\substack{\text{inlet}\\\text{port}}}.$$

We further use the following convention to index outlet ports of subsystem $N + 1$ (the raw material mixer):

$$h(i, N) = (N + 1, j^*)|_{j^*=i}, \forall i = 1, ..., N. \tag{3.2}$$

https://doi.org/10.51202/9783186952035 - Generiert durch IP 216.73.216.36, am 18.01.2026, 21:11:22. © Urheberrechtlich geschützter Inhalt. Ohne gesonderte Erlaubnis ist jede urheberrechtliche Nutzung untersagt, insbesondere die Nutzung des Inhalts im Zusammenhang mit, für oder in KI-Systemen, KI-Modellen oder Generativen Sprachmodellen.

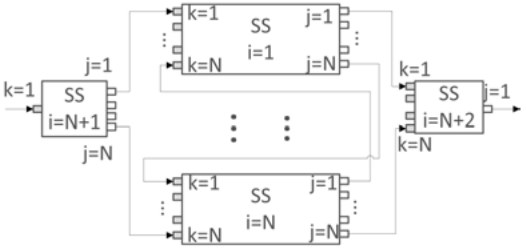


Figure 3.2: Subsystems, ports and connections in an open-loop N-reactor network superstructure. SS is short for subsystem. i is an index for a subsystem. j is an index for an outlet port. k is an index for an inlet port.

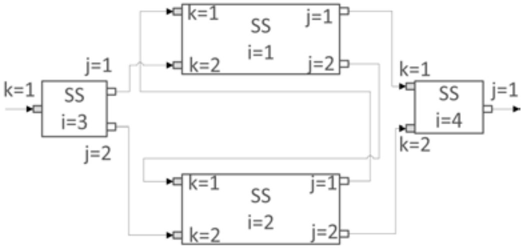


Figure 3.3: Subsystems, ports and connections in an exemplary open-loop 2-reactor network.

Hence, the inlet port (i, N) of the i-th reactor is always connected to the i-th outlet port of subsystem $N + 1$.

For an outlet (inlet) port, we sometimes do not need to indicate the index of its connected inlet (outlet) port but to indicate the index of its connected subsystem (reactor). Therefore, we introduce functions $\bar{l}(i, j) = i'$ and $\bar{h}(i', k') = i$. In particular, $\bar{l}(i, j)$ indicates subsystem i', whose k'-th inlet port is connected to outlet port (i, j). Similarly, $\bar{h}(i', k')$ indicates subsystem i, whose j-th outlet port is connected to inlet port (i, j). Table 3.2 gives a summary of the used symbols.

Example (continued)**.** *Subsystems, ports and connections for an exemplary network consisting of 2 reactors are shown in Fig. 3.3. Subsystems 1 and 2 refer to reactors, subsystem 3 refers to a splitter splitting the raw material feed, while subsystem 4 refers to the product mixer. For every subsystem, its outlet ports and inlet ports are labeled by* j *and* k *in the figure, respectively. For example, subsystem 1 has two outlet ports* $(i = 1, j = 1)$ *and* $(i = 1, j = 2)$*. Subsystem 1 also has two inlet ports* $(i = 1, k = 1)$ *and* $(i = 1, k = 2)$*. We know that, e.g.,* $l(i = 1, j = 2) = (i = 2, k = 1)$ *and* $h(i = 2, k = 2) = (i = 3, j = 2)$*.* $\bar{l}(1, 2) = 2$ *and* $\bar{h}(2, 2) = 3$*.* □

https://doi.org/10.51202/9783186952035 - Generiert durch IP 216.73.216.36, am 18.01.2026, 21:11:22. © Urheberrechtlich geschützter Inhalt. Ohne gesonderte Erlaubnis ist jede urheberrechtliche Nutzung untersagt, insbesondere die Nutzung des Inhalts im Zusammenhang mit, für oder in KI-Systemen, KI-Modellen oder Generativen Sprachmodellen.

Table 3.2: Used symbols for modeling the reactor network.

symbols	meaning
i	a subsystem
j	an outlet port
k	an inlet port
(i,j)	j-th outlet port of subsystem i
(i,k)	k-th inlet port of subsystem i
$l(i,j)$	an inlet port, connected to (i,j)
$h(i,k)$	an outlet port, connected to (i,k)
$(i,j) \triangleright (i',k')$	a pipe connection from (i,j) to (i',k')
$\bar{l}(i,j)$	a subsystem, one of its inlets is connected to (i,j)
$\bar{h}(i,k)$	a subsystem, one of its outlets is connected to (i,k)

3.1.2 Models of Subsystem 1 to N

In this subsection we set up the model for subsystems, which correspond to a reactor. We limit the presentation to networks with only CSTR in the following and refer to Appendix A for an extension to also include PFR. Denote the number of chemical components necessary to describe the reactions in a reactor by N_c [1]. If we use $x_i \in \mathbb{R}^{N_c+1}$ to represent the N_c concentration in $[mol/m^3]$ of all modeled components and the temperature in $[K]$ inside the reactor and if we use $u_{i,k} \in \mathbb{R}^{N_c+1}$, $k = 1, \cdots, N$, to represent N_c component flowrates in $[mol/s]$ and the energy flowrate in $[J/s]$ through inlet port (i,k), subsystem i, $i \in \{1, ..., N\}$, can be modeled by

$$\dot{x}_i = f_i(x_i, u_{i,1}, \cdots, u_{i,N}, q_{i,1}, \cdots, q_{i,N}, p_i), x_i(0) = x_i^0. \tag{3.3}$$

$q_{i,j} \in \mathbb{R}$, $j = 1, \cdots, N$, is a positive scalar variable representing the volumetric flowrate in $[m^3/s]$ of the material getting out of outlet port (i,j). The indices of q always refer to outlet ports (we do not introduce variables $q_{i,k}$, in which (i,k) refers to an inlet port). p_i is a vector of design parameters for reactor i, including for example the reactor volume or its pressure. $f_i(\cdot) \in \mathbb{C}^\infty$ is a smooth function resulting from mass and energy balances. x_i^0 denotes initial conditions of x_i. In terms of systems theory, x_i are the state variables of subsystem i, while $u_{i,k}$ are the inputs of subsystem i. $q_{i,j}$ and p_i are design parameters of subsystem i.

We use $y_{i,j} \in \mathbb{R}^{N_c+1}$ to denote the molar flowrates of each components in $[mol/s]$ and the energy flowrate in $[J/s]$ through outlet port (i,j). $y_{i,j}$ has a dimension of $N_c + 1$. Now we present a fundamental trick, which will be used throughout this chapter to facilitate the eigenvalue-based analysis and problem reformulation below. We propose that $y_{i,j}$ can be modeled by

$$y_{i,j} = q_{i,j} g_{i,j}(x_i, p_i), \forall j = 1, \cdots, N. \tag{3.4}$$

$q_{i,j}$ has been already defined as volumetric flowrate in $[m^3/s]$ through outlet port (i,j). $g_{i,j}(\cdot) \in \mathbb{C}^\infty$ is a vector-valued smooth function of x_i and p_i. Because the dimension and

---

[1] It is typically not necessary to model concentrations of all components in a reactor. For example, in the allyl chloride example, only the concentrations of A, B, C are needed to model the reactor, while the concentrations of other side products can be determined from mass balances. For this reason, we can choose $N_c = 3$ for the allyl chloride example.

https://doi.org/10.51202/9783186952035 - Generiert durch IP 216.73.216.36, am 18.01.2026, 21:11:22. © Urheberrechtlich geschützter Inhalt. Ohne gesonderte Erlaubnis ist jede urheberrechtliche Nutzung untersagt, insbesondere die Nutzung des Inhalts im Zusammenhang mit, für oder in KI-Systemen, KI-Modellen oder Generativen Sprachmodellen.

elements of $y_{i,j}$ and $q_{i,j}$ have already been defined, $g_{i,j}(x_i, p_i)$ should be of dimension N_c+1. The first N_c elements of $g_{i,j}(x_i, p_i)$ refer to molar densities in $[mol/m^3]$ of each component through outlet port (i, j) and another element refers to the energy density in $[J/m^3]$ of the flow through outlet port (i, j). In systems theory, $y_{i,j}$ is called an output of subsystem i.

It is important to stress the multiplication on the right hand side of Eq. (3.4). From a physical point of view, $q_{i,j}$ can be viewed as "valve position" which sets the flowrate in pipe $(i, j) \triangleright l(i, j)$. Hence, if $q_{i,j} = 0$, then $y_{i,j} = 0$ according to Eq. (3.4). The valve is closed and there is no material passing through the pipe. If $q_{i,j} > 0$, the valve is open and $y_{i,j}$ represents the component and energy flows leaving reactor i through the pipe. This way, $q_{i,j}$ is used to also make decisions on the existence of connections (pipes) in the superstructure without introducing any discrete variables.

Note that the input vectors $u_{i,k}$ and output vectors $y_{i,j}$ are of the same type to facilitate the realization of connections between every pair of outlet and inlet ports (i, j) and $l(i, j)$, respectively.

Example (continued)**.** *We consider a superstructure of the 2-reactor network in Fig. 3.3. Each reactor is assumed to be a CSTR. The mass balance for component A in reactor 1 is*

$$V_1 \dot{c}_{A1} = \dot{n}^0_{A1,1} + \dot{n}^0_{A1,2} - (q_{1,1} + q_{1,2}) c_{A1} + V_1 R_{A1}. \tag{3.5}$$

c_{A1} *denotes molar concentration in [mol/m*^{3}*] of A inside reactor 1.* $\dot{n}^0_{A1,1}$ *and* $\dot{n}^0_{A1,2}$ *denote molar flowrates in [mol/s] of A entering the reactor through inlet ports* $(1, 1)$ *and* $(1, 2)$*, respectively.* $q_{1,1}$ *and* $q_{1,2}$ *are volumetric flowrates in [m*^{3}*/s] leaving the reactor through outlet ports* $(1, 1)$ *and* $(1, 2)$*, respectively.* V_1 *denotes the reactor volume in [m*^{3}*] and* R_{A1} *the reaction rate [mol/m*^{3}*/s] of A. Similarly, we can formulate the mass balances for B and C as*

$$V_1 \dot{c}_{B1} = \dot{n}^0_{B1,1} + \dot{n}^0_{B1,2} - (q_{1,1} + q_{1,2}) c_{B1} + V_1 R_{B1}, \tag{3.6}$$

$$V_1 \dot{c}_{C1} = \dot{n}^0_{C1,1} + \dot{n}^0_{C1,2} - (q_{1,1} + q_{1,2}) c_{C1} + V_1 R_{C1}. \tag{3.7}$$

The energy balance for reactor 1 is given by

$$c_p V_1 \dot{T}_1 = \dot{Q}^0_{1,1} + \dot{Q}^0_{1,2} - (q_{1,1} + q_{1,2}) c_p T_1 + V_1 \sum_{i=1,2,3} H_i r_i + Q_h. \tag{3.8}$$

T_1 *denotes the temperature in [K] in reactor 1,* c_p *the volumetric heat capacity in [J/m*^{3}*/K],* $\dot{Q}^0_{1,1}$ *and* $\dot{Q}^0_{1,2}$ *the energy flowrates in [J/s] entering the reactor through inlet ports* $(1, 1)$ *and* $(1, 2)$*, respectively, and* H_1*,* H_2 *and* H_3 *the heats of reaction in [J/mol] (refer to Table 3.1).* r_1*,* r_2 *and* r_3 *are defined in Eq. (3.1).* Q_h *denotes energy duty in [J/s] of the reactor's heating or cooling jacket.*

Reactor 1 has two outlets, which can be modeled by

$$y_{1,1} = (\underbrace{q_{1,1} c_{A1},\ q_{1,1} c_{B1},\ q_{1,1} c_{C1},}_{\substack{\text{molar flowrates in [mol/s]} \\ \text{through outlet} \\ \text{port } (1,1)}}\ \underbrace{q_{1,1} c_p T_1}_{\substack{\text{energy flowrate in} \\ \text{[J/s] through} \\ \text{outlet port } (1,1)}})^T, \tag{3.9}$$

and similarly

$$y_{1,2} = (q_{1,2} c_{A1},\ q_{1,2} c_{B1},\ q_{1,2} c_{C1},\ q_{1,2} c_p T_1)^T. \tag{3.10}$$

https://doi.org/10.51202/9783186952035 - Generiert durch IP 216.73.216.36, am 18.01.2026, 21:11:22. © Urheberrechtlich geschützter Inhalt. Ohne gesonderte Erlaubnis ist jede urheberrechtliche Nutzung untersagt, insbesondere die Nutzung des Inhalts im Zusammenhang mit, für oder in KI-Systemen, KI-Modellen oder Generativen Sprachmodellen.

If we use the abbreviations

$$\begin{aligned} x_1 &:= (c_{A1}, c_{B1}, c_{C1}, T)^T, \\ u_{1,1} &:= (\dot{n}^0_{A1,1}, \dot{n}^0_{B1,1}, \dot{n}^0_{C1,1}, \dot{Q}^0_{1,1})^T, \\ u_{1,2} &:= (\dot{n}^0_{A1,2}, \dot{n}^0_{B1,2}, \dot{n}^0_{C1,2}, \dot{Q}^0_{1,2})^T, \\ p_1 &:= V_1, \end{aligned}$$

Eqs. (3.5)-(3.8) can be generally written by Eq. (3.3). If we furthermore define

$$\begin{aligned} g_{1,1}(x_1, p_1) &:= (c_{A1}, c_{B1}, c_{C1}, c_p T_1)^T, \\ g_{1,2}(x_1, p_1) &:= (c_{A1}, c_{B1}, c_{C1}, c_p T_1)^T, \end{aligned}$$

Eqs. (3.9), (3.10) have the same structure as Eq. (3.4).

Obviously, we can write exactly the same kind of equations for reactor 2. To this end, we have shown that Eqs. (3.3), (3.4) constitute an abstracted form of the subsystem models in a 2*-reactor network.*

We discuss shortly the role of $q_{1,1}$ *and* $q_{1,2}$ *in Eqs. (3.9), (3.10). When* $q_{1,1} = 0$*,* $y_{1,1} = 0$*, the connection* $(1,1) \triangleright (4,1)$ *does not carry any material flow. When* $q_{1,1} > 0$*,* $y_{1,1} \geq 0$*, there is a material and energy flow leaving the reactor through connection* $(1,1) \triangleright (4,1)$*. Thus* $q_{1,1}$ *determines whether the connection* $(1,1) \triangleright (4,1)$ *exists in the superstructure or not. Similar interpretations hold for all other flowrate variables. This way, we have represented structural alternatives of existent and non-existent connections by means of continuous flowrate variables* $q_{i,j}$*. No binary variables have been introduced so far.* ☐

3.1.3 Models of subsystems $N+1$ and $N+2$

There are two units, subsystem $N+1$ and $N+2$, which have not been modeled so far. Subsystem $N+1$ represents the raw material splitter, which can be modeled by the algebraic equations

$$y_{N+1,j} = q_{N+1,j} p_{sys}, \forall j = 1, \cdots, N. \tag{3.11}$$

$p_{sys} \in \mathbb{R}^{N_c+1}$ denotes molar concentration in $[mol/m^3]$ and the energy density in $[J/m^3]$ in the feed. Hence, p_{sys} is of dimension $N_c + 1$. $q_{N+1,j}$, $j = 1, \cdots, N$, is a scalar variable representing the volumetric flowrates in $[m^3/s]$ through outlet port $(N+1, j)$. Again, $q_{N+1,j}$ can be interpreted also as "valve positions". If $q_{N+1,j} = 0$, the valve for connection $(N+1, j) \triangleright l(N+1, j)$ is fully closed, while $q_{N+1,j} > 0$ means that, the valve is open. $y_{N+1,j}$ represent the molar flowrates in $[mol/s]$ and the energy flowrate in $[J/s]$ through outlet port $(N+1, j)$.

Subsystem $N+2$ is a mixer, which generates the product flow. It can be modeled by

$$y_{sys} = \sum_{k=1,\cdots,N} u_{N+2,k} \tag{3.12}$$

with $u_{N+2,k} \in \mathbb{R}^{N_c+1}$ having the same dimension and the same type of elements as other $u_{i,k}$ presented before. $u_{N+2,k}$ represents the molar flowrates in $[mol/s]$ of each components and the energy flowrate in $[J/s]$ through inlet port $(N+2, k)$. $y_{sys} \in \mathbb{R}^{N_c+1}$ has the same dimension and the same type of elements as the other $y_{i,j}$ presented before.

https://doi.org/10.51202/9783186952035 - Generiert durch IP 216.73.216.36, am 18.01.2026, 21:11:22. © Urheberrechtlich geschützter Inhalt. Ohne gesonderte Erlaubnis ist jede urheberrechtliche Nutzung untersagt, insbesondere die Nutzung des Inhalts im Zusammenhang mit, für oder in KI-Systemen, KI-Modellen oder Generativen Sprachmodellen.

Example (continued). *Subsystem 3 in the 2-reactor network can be modeled by*

$$y_{3,1} = q_{3,1}(c_A^{sys}, c_B^{sys}, c_C^{sys}, E^{sys})^T, \tag{3.13}$$

$$y_{3,2} = q_{3,2}(c_A^{sys}, c_B^{sys}, c_C^{sys}, E^{sys})^T. \tag{3.14}$$

c_A^{sys}*,* c_B^{sys}*,* c_C^{sys} *denote concentrations in [mol/m*^{3}*] of components A, B, C and* E^{sys} *the energy density in [J/m*^{3}*] in the feed to the reactor network.* $q_{3,1}$ *and* $q_{3,2}$ *are the material streams in [m*^{3}*/s] through outlet ports* $(3,1)$ *and* $(3,2)$*, respectively.* $y_{3,1}$ *and* $y_{3,2}$ *represent the molar flowrates of each components and the energy density flowrate through these outlet ports.*

The output y_{sys} *of the product mixer can be formulated as*

$$y_{sys} = \begin{bmatrix} \dot{n}^0_{A4,1} + \dot{n}^0_{A4,2} \\ \dot{n}^0_{B4,1} + \dot{n}^0_{B4,2} \\ \dot{n}^0_{C4,1} + \dot{n}^0_{C4,2} \\ \dot{Q}^0_{4,1} + \dot{Q}^0_{4,2} \end{bmatrix} \tag{3.15}$$

with the molar flowrates $\dot{n}^0_{A4,1}$*,* $\dot{n}^0_{B4,1}$*,* $\dot{n}^0_{C4,1}$*,* $\dot{n}^0_{A4,2}$*,* $\dot{n}^0_{B4,2}$*,* $\dot{n}^0_{C4,2}$ *of components A, B, C in [mol/s] and the energy flowrates* $\dot{Q}^0_{4,1}$*,* $\dot{Q}^0_{4,2}$ *in [J/s] through inlet ports* $(4,1)$ *and* $(4,2)$*, respectively.* y_{sys} *denotes the molar component flowrates and the energy density flowrate through outlet port* $(4,1)$*.*

If we introduce

$$p_{sys} := (c_A^{sys}, c_B^{sys}, c_C^{sys}, E^{sys})^T, \tag{3.16}$$

Eqs. (3.13), (3.14) have the same structure as Eq. (3.11). If furthermore

$$u_{4,1} := (\dot{n}^0_{A4,1}, \dot{n}^0_{B4,1}, \dot{n}^0_{C4,1}, \dot{Q}_{4,1})^T,$$
$$u_{4,2} := (\dot{n}^0_{A4,2}, \dot{n}^0_{B4,2}, \dot{n}^0_{C4,2}, \dot{Q}_{4,2})^T,$$

are defined as inputs of subsystem 4, the model of subsystem 4, Eq. (3.15), has the same form as Eq. (3.12). □

3.1.4 Modeling flow connections

Having introduced the models for the individual subsystems, $i = 1, \cdots, N+2$, we need to specify connections to link subsystems. In the superstructure, each inlet port (i,k) is connected to an outlet port $h(i,k)$ through connection $h(i,k) \triangleright (i,k)$. Because $u_{i,k}$ and $y_{h(i,k)}$ are of the same type, connections are simply given by the equations

$$u_{i,k} = y_{h(i,k)}, \forall i = 1, \cdots, N, \forall k = 1, \cdots, N, \tag{3.17}$$

$$u_{(N+2,k)} = y_{h(N+2,k)}, \forall k = 1, \cdots, N. \tag{3.18}$$

Eqs. (3.17), (3.18) connect all inlet ports of subsystems 1, $\cdots$, N and $N+2$. Subsystem $N+1$ does not need to be connected, because it only has the feed to the reactor network as its input.

Example (continued). *For the 2-reactor network example considered before, the following connections hold:*

$$\begin{aligned} &u_{1,1} = y_{2,1}, \; u_{1,2} = y_{3,1}, \\ &u_{2,1} = y_{1,2}, \; u_{2,2} = y_{3,2}, \\ &u_{4,1} = y_{1,1}, \; u_{4,2} = y_{2,2}. \end{aligned} \tag{3.19}$$

□

https://doi.org/10.51202/9783186952035 - Generiert durch IP 216.73.216.36, am 18.01.2026, 21:11:22. © Urheberrechtlich geschützter Inhalt. Ohne gesonderte Erlaubnis ist jede urheberrechtliche Nutzung untersagt, insbesondere die Nutzung des Inhalts im Zusammenhang mit, für oder in KI-Systemen, KI-Modellen oder Generativen Sprachmodellen.

3.1.5 A dynamic model of the network

After introducing individual models for subsystems and their connections, we are ready to formulate a dynamic model for the open-loop N-reactor network superstructure. To get a compact form, we eliminate all internal $u_{i,k}$ and $y_{i,j}$ by replacing $y_{i,j}$ in Eqs. (3.17), (3.18) by Eqs. (3.4), (3.11) and all $u_{i,k}$ in Eqs. (3.3), (3.12) by the resulting equations from the previous step, which leads to

$$\begin{aligned} \dot{x}_i = f_i(x_i,\ & \underbrace{\cdots,\ q_{h(i,k)} g_{h(i,k)}(x_{\bar{h}(i,k)}, p_{\bar{h}(i,k)}),\ \cdots}_{k=1,\cdots,N-1}, \\ & q_{N+1,i} p_{sys},\ q_{i,1},\ \cdots,\ q_{i,N},\ p_i),\ \forall i = 1, \cdots, N. \end{aligned} \tag{3.20}$$

Note that, the indexing convention shown in Eq. (3.2) has been used to derive the term $q_{N+1,i} p_{sys}$ in the above equation. That is, the N-th inlet port of subsystem i is connected to the outlet port i-th outlet port of the raw material splitter.

After eliminating $u_{N+2,k}$ in Eq.(3.12), the output y_{sys} of the reactor network is

$$y_{sys} = \sum_{k=1,\cdots,N} q_{h(N+2,k)} g_{h(N+2,k)}(x_{\bar{h}(N+2,k)}, p_{\bar{h}(N+2,k)}). \tag{3.21}$$

Eq. (3.20) constitutes the state equations of the N-reactor network, while Eq. (3.21) refers to the output equation. An important feature of the structure of Eq. (3.20) is that the values of $q_{i,j}$ determine the existence of connections in the superstructure without introducing integer variables. The model (3.20), (3.21) contains variables summarized in Table 3.3. The degrees of freedom, denoted as ψ_o, are given by q, p, and p_{sys}, i.e.,

$$\psi_o = (q^T, p^T, p_{sys}^T)^T. \tag{3.22}$$

Table 3.3: State variables and design parameters in Eqs. (3.20), (3.21).

state variables:	$x := (x_1^T, \cdots, x_N^T)^T \in \mathbb{R}^{\sum_i n_{x_i}}$.
design parameters:	$q := (q_{1,1}, \cdots, q_{N+1,N})^T \in \mathbb{R}^{N(N+1)}$,
	$p := (p_1^T, \cdots, p_N^T)^T \in \mathbb{R}^{\sum_i n_{p_i}}$,
	$p_{sys} \in \mathbb{R}^{N_c+1}$.
outputs variables:	$y_{sys} \in \mathbb{R}^{N_c+1}$.

Example (continued). *After eliminating the internal variables the following model for the 2-reactor network is obtained:*

$$\begin{aligned} V_1 \dot{c}_{A1} &= q_{3,1}[p_{sys}]_1 + q_{2,1} c_{A2} - (q_{1,1} + q_{1,2}) c_{A1} + V_1 R_{A1}, \\ V_1 \dot{c}_{B1} &= q_{3,1}[p_{sys}]_2 + q_{2,1} c_{B2} - (q_{1,1} + q_{1,2}) c_{B1} + V_1 R_{B1}, \\ V_1 \dot{c}_{C1} &= q_{3,1}[p_{sys}]_3 + q_{2,1} c_{C2} - (q_{1,1} + q_{1,2}) c_{C1} + V_1 R_{C1}, \\ c_p V_1 \dot{T}_1 &= q_{3,1}[p_{sys}]_4 + q_{2,1} c_p T_2 - (q_{1,1} + q_{1,2}) c_p T_1 + V_1 \sum_{i=1,2,3} H_i r_i + Q_{h1}, \\ V_2 \dot{c}_{A2} &= q_{3,2}[p_{sys}]_1 + q_{1,2} c_{A1} - (q_{2,1} + q_{2,2}) c_{A2} + V_2 R_{A2}, \\ V_2 \dot{c}_{B2} &= q_{3,2}[p_{sys}]_2 + q_{1,2} c_{B1} - (q_{2,1} + q_{2,2}) c_{B2} + V_2 R_{B2}, \\ V_2 \dot{c}_{C2} &= q_{3,2}[p_{sys}]_3 + q_{1,2} c_{C1} - (q_{2,1} + q_{2,2}) c_{C2} + V_2 R_{C2}, \\ c_p V_2 \dot{T}_2 &= q_{3,2}[p_{sys}]_4 + q_{1,2} c_p T_1 - (q_{2,1} + q_{2,2}) c_p T_2 + V_2 \sum_{i=1,2,3} H_i r_i + Q_{h2}, \end{aligned} \tag{3.23}$$

https://doi.org/10.51202/9783186952035 - Generiert durch IP 216.73.216.36, am 18.01.2026, 21:11:22. © Urheberrechtlich geschützter Inhalt. Ohne gesonderte Erlaubnis ist jede urheberrechtliche Nutzung untersagt, insbesondere die Nutzung des Inhalts im Zusammenhang mit, für oder in KI-Systemen, KI-Modellen oder Generativen Sprachmodellen.

$[p_{sys}]_l$, $l = 1, \cdots, 4$, *denotes the l-th element in vector* p_{sys}, *according to Eq. (3.16).* □

3.1.6 Idle reactors in open-loop reactor networks

Next we discuss the concept of idle reactors in open-loop reactor networks and formalize its definition. The open-loop superstructure in Fig. 3.1 comprises a sufficiently large number of fully connected reactors. It is the objective of optimization to determine how many and which reactors are kept in the final design. Reactors in the superstructure are called non-idle, if they are included in the optimal flowsheet. Otherwise, they are called idle reactors.

An idle reactor i can be realized in the model by setting flowrate variables $q_{i,j} = 0$, $\forall j = 1, \cdots, N$, and $q_{h(i,k)} = 0$, $\forall k = 1, \cdots, N$. It physically represents the case that all inlet and outlet ports are closed. Because material is neither getting in nor getting out of reactor i in this case, an idle reactor i does not influence the other reactors in the flowsheet. We can formalize the definition of idle reactors for the purpose of open-loop reactor network synthesis as follows:

Definition 3.1.1 (Idle reactor in an open-loop reactor network)**.** *A subsystem* i, $i = 1, \cdots, N$, *is an idle reactor, if there is neither material getting into nor material getting out of the reactor. That is*

$$\begin{aligned} q_{(i,j)} &= 0, \ \forall j = 1, \cdots, N, \\ q_{h(i,k)} &= 0, \ \forall k = 1, \cdots, N. \end{aligned} \tag{3.24}$$

Otherwise, the reactor is called non-idle.

The definitions of idle reactors used throughout this chapter is illustrated by the example considered before.

Example (continued)**.** *If reactor 1 in Fig. 3.3 is idle, Eq. (3.24) becomes*

$$q_{3,1} = q_{2,1} = q_{1,1} = q_{1,2} = 0, \tag{3.25}$$

to represent closing all inlet and outlet ports of reactor 1. Inserting Eq. (3.25) into Eq. (3.23), the model of the idle reactor 1 becomes

$$\begin{aligned} \dot{c}_{A1} &= R_{A1}, \\ \dot{c}_{B1} &= R_{B1}, \\ \dot{c}_{C1} &= R_{C1}, \\ c_p V_1 \dot{T}_1 &= V_1 \sum_{i=1,2,3} H_i r_i + Q_h, \end{aligned} \tag{3.26}$$

which is in fact a batch reactor model. If reactor 2 is non-idle, namely if

$$q_{3,2} > 0 \ \ or \ \ q_{2,2} > 0,$$

the model of reactor 2 becomes

$$\begin{aligned} V_2 \dot{c}_{A2} &= q_{3,2}[p_{sys}]_1 - q_{2,2} c_{A2} + V_2 R_{A2}, \\ V_2 \dot{c}_{B2} &= q_{3,2}[p_{sys}]_2 - q_{2,2} c_{B2} + V_2 R_{B2}, \\ V_2 \dot{c}_{C2} &= q_{3,2}[p_{sys}]_3 - q_{2,2} c_{C2} + V_2 R_{C2}, \\ c_p V_2 \dot{T}_2 &= q_{3,2}[p_{sys}]_4 - q_{2,2} c_p T_2 + V_2 \sum_{i=1}^{3} H_i r_i + Q_{h2}. \end{aligned} \tag{3.27}$$

https://doi.org/10.51202/9783186952035 - Generiert durch IP 216.73.216.36, am 18.01.2026, 21:11:22. © Urheberrechtlich geschützter Inhalt. Ohne gesonderte Erlaubnis ist jede urheberrechtliche Nutzung untersagt, insbesondere die Nutzung des Inhalts im Zusammenhang mit, für oder in KI-Systemen, KI-Modellen oder Generativen Sprachmodellen.

Obviously, model (3.26) of the idle reactor 1 is independent of model (3.27) of the non-idle reactor 2. If we would increase the values of $q_{3,1}$, $q_{2,1}$, $q_{1,1}$, or $q_{1,2}$ to a positive number, reactor 1 will be activated. This leads to a discontinuity of the opeo-loop reactor network model and its Jacobian matrix, which is of great significance for formulating an optimization problem in the next section. ☐

3.2 Problem formulation

3.2.1 Eigenvalue constraint for open-loop reactor network synthesis

Although the network model (3.20) is a specialized form of the ODE system (2.1), for design purposes we are not interested in the eigenvalue-based dynamic properties (i.e. stability and response speed, refer to Chapter 2) of the whole network, but only interested in the dynamic properties of interconnected *non-idle* reactors in the network. In this subsection, we will present an eigenvalue constraint only for non-idle reactors, which will be used in the problem formulation.

A reactor i is idle, if the flowrates $q_{i,j} = 0, \forall\ j = 1, \cdots, N$, and $q_{h(i,k)} = 0, \forall\ k = 1, \cdots, N$. Hence, idle and non-idle reactors can be easily distinguished by inspection of q. Let $\mathcal{I}$ be the index set of all reactors, $\mathcal{I}_{id}(q)$ the index set for idle reactors and $\mathcal{I}_{nid}(q)$ the index set for non-idle reactors, which are determined from inspecting the values of q. That is,

$$\begin{aligned} \mathcal{I} &= \{i \mid i = 1, \cdots, N\} \\ \mathcal{I}_{id}(q) &= \{i \in \mathcal{I} \mid \text{reactor } i \text{ is idle}\}, \\ \mathcal{I}_{nid}(q) &= \{i \in \mathcal{I} \mid \text{reactor } i \text{ is non-idle}\}. \end{aligned} \tag{3.28}$$

Because the cases where all reactors are either idle or non-idle are trivial, throughout this paper, without explicitly mentioning, we assume that

$$\mathcal{I}_{id} \neq \emptyset \text{ and } \mathcal{I}_{nid} \neq \emptyset. \tag{3.29}$$

The open-loop network model (3.20) can now be split into a submodel with only idle reactors and another submodel with any non-idle reactors. The submodel of idle reactors is

$$\dot{x}_i = \quad f_i(x_i, \underbrace{0, \cdots, 0}_{u_{i,k}=0}, \underbrace{0, \cdots, 0}_{q_{i,j}=0}, p_i), \forall i \in \mathcal{I}_{id}(q). \tag{3.30}$$

The submodel of non-idle reactors is

$$\begin{aligned} \dot{x}_i = f_i(x_i, & \underbrace{\cdots, q_{h(i,k)} g_{h(i,k)}(\cdot), \cdots}_{\substack{\text{if } u_{i,k} \text{ is connected} \\ \text{connected to} \\ \text{non-idle rectors}}}, \underbrace{\cdots, 0, \cdots}_{\substack{\text{if } u_{i,k} \text{ is} \\ \text{connected to} \\ \text{idle reactors}}}, \underbrace{q_{N+1,i} p_{sys}}_{\text{system feed}}, \\ & \underbrace{\cdots, q_{i,k}, \cdots}_{\substack{\text{if } q_{i,j} \text{ feeds} \\ \text{into non-idle} \\ \text{reactors}}}, \underbrace{\cdots, 0, \cdots}_{\substack{\text{if } q_{i,j} \text{ feeds} \\ \text{into idle reactors}}}, p_i), \forall i \in \mathcal{I}_{nid}(q). \end{aligned} \tag{3.31}$$

https://doi.org/10.51202/9783186952035 - Generiert durch IP 216.73.216.36, am 18.01.2026, 21:11:22. © Urheberrechtlich geschützter Inhalt. Ohne gesonderte Erlaubnis ist jede urheberrechtliche Nutzung untersagt, insbesondere die Nutzung des Inhalts im Zusammenhang mit, für oder in KI-Systemen, KI-Modellen oder Generativen Sprachmodellen.

Note that, as already seen in the Eqs. (3.26), (3.27), the submodel of non-idle reactors (3.31) is independent of the submodel of idle-reactors (3.30). This is a useful consequence of the modeling approach which represents the output $y_{i,j}$ as a product of the flowrate variables $q_{i,j}$ and the vector $g_{i,j}(\cdot)$ in Eqs. (3.4), (3.11). This feature of the submodels results later in a structured Jacobian matrix of the open-loop reactor network. The inner submatrices change if a reactor transitions from idle to non-idle mode or vice versa, as discussed in more detail below.

Let us use $J_{tot}(x, q, p, p_{sys})$ to denote the Jacobian matrix of the open-loop reactor network model (3.20), $J_{id}(x, q, p, p_{sys})$ and $J_{nid}(x, q, p, p_{sys})$ to denote the Jacobian matrices of the submodel (3.30) for idle reactors and the submodel (3.31) for non-idle reactors, respectively. Because the submodel of idle reactors and the submodel of non-idle reactors are independent to each other, with probably reordering the sequence of subsystems, we have

$$J_{tot} = \begin{bmatrix} J_{id} & 0 \\ 0 & J_{nid} \end{bmatrix}. \tag{3.32}$$

Reactor network design with guaranteed eigenvalue-based properties should only consider the eigenvalues of J_{nid} of non-idle reactors, but not the ones of J_{tot} of all reactors in the network. So an eigenvalue constraint for guaranteeing dynamic properties of the open-loop reactor network, i.e., stability and response speed (cf. Chapter 2), can be formulated as

$$\alpha_{J_{nid}}(x, q, p, p_{sys}) < -c, \tag{3.33}$$

in which $\alpha_{J_{nid}}(\cdot)$ denotes the spectral abscissa of matrix $J_{nid}(x, q, p, p_{sys})$. $c > 0$ is a given constant. Note that $\alpha_{J_{nid}}(\cdot)$ is defined not only for the steady states of the dynamic system (3.20). $D_o := \mathbb{R}^{n_x} \times \mathbb{R}^{n_q} \times \mathbb{R}^{n_p} \times \mathbb{R}^{n_{p_{sys}}} \backslash \{(x^T, 0^T, p^T, p_{sys}^T)^T\}$ is the domain of the function $\alpha_{J_{nid}}(\cdot)$. Trivial points $(x^T, q^T, p^T, p_{sys}^T)^T$ with $q = 0$ are excluded, because they refer to an empty J_{nid} (all reactors are idle). Hence, Eq. (3.33) is a well-defined constraint for numerical optimization, although, as it will be seen in the next subsection, this constraint is not continuous everywhere.

3.2.2 Continuity analysis of the proposed eigenvalue constraint

Before we analyze the continuity of the constructed constraint (3.33), we briefly review the continuity property of the SA function $\alpha_{J_{tot}}(\cdot)$ of matrix $J_{tot}(\cdot)$ (cf. Section 2.2). A short conclusion is that, because all elements in $J_{tot}(x, q, p, p_{sys})$ are continuous functions of its arguments, $\alpha_{J_{tot}}(x, q, p, p_{sys})$ is a continuous function (not necessarily smooth) from $\mathbb{R}^{n_x} \times \mathbb{R}^{n_q} \times \mathbb{R}^{n_p} \times \mathbb{R}^{n_{p_{sys}}}$ to $\mathbb{R}$.

Although $\alpha_{J_{tot}}(\cdot)$ is continuous, $\alpha_{J_{nid}}(\cdot)$ is, however, in general *discontinuous.* This is because $J_{tot}(\cdot)$ is a matrix of a fixed dimension, while $J_{nid}(\cdot)$ may change its size depending on different values of q. When an idle reactor i is activated by changing some values of the flowrates q, J_{nid} increases its size. Here, we give a condition to check the continuity of constraint (3.33).

Proposition 3.2.1 (Continuity of $\alpha_{J_{nid}}(\cdot)$)**.** *If* $J_{tot}(x, q, p, p_{sys}) \in \mathbb{R}^{n_x} \times \mathbb{R}^{n_q} \times \mathbb{R}^{n_p} \times \mathbb{R}^{n_{p_{sys}}} \rightarrow \mathbb{R}^{n_x \times n_x}$ *is a continuous function of* x*,* q*,* p *and* p_{sys}*, assume that Eq. (3.29) holds at a point* $(x^T, q^T, p^T, p_{sys}^T)^T$*,* $\alpha_{J_{nid}}(x, q, p, p_{sys})$ *is locally continuous at this point,* iff

$$\alpha_{J_{id}}(x, q, p, p_{sys}) \leq \alpha_{J_{nid}}(x, q, p, p_{sys}). \tag{3.34}$$

https://doi.org/10.51202/9783186952035 - Generiert durch IP 216.73.216.36, am 18.01.2026, 21:11:22. © Urheberrechtlich geschützter Inhalt. Ohne gesonderte Erlaubnis ist jede urheberrechtliche Nutzung untersagt, insbesondere die Nutzung des Inhalts im Zusammenhang mit, für oder in KI-Systemen, KI-Modellen oder Generativen Sprachmodellen.

Table 3.4: A steady state P^* of the 2-reactor network example in Fig. 3.3. The system is fed with 10 mol/s A and 10 mol/s C at temperature 300 K. $q_{3,1}[p_{sys}]_1$, $q_{3,1}[p_{sys}]_2$ and $q_{3,1}[p_{sys}]_3$ denote molar flowrates of components A, B, C, and $q_{3,1}[p_{sys}]_4$ the energy flowrate through port $(3,1)$. Likewise, $q_{3,2}[p_{sys}]_1$, $q_{3,2}[p_{sys}]_2$ and $q_{3,2}[p_{sys}]_3$ denote molar flowrates of components A, B, C, and $q_{3,2}[p_{sys}]_4$ the energy flowrate through port $(3,2)$.

variable	value at P^*	units
c_1	$(0, 0, 1/22.4)^T$	$[mol/l]$
c_2	$(0.2246, 0.0035, 0.2246)^T$	$[mol/l]$
T_1	450	$[K]$
T_2	450	$[K]$
V_1	100	$[l]$
V_2	100	$[l]$
Q_{h1}	0	$[MJ/s]$
Q_{h2}	-1.043	$[MJ/s]$
$q_{1,1}$	0	$[g/s]$
$q_{1,2}$	0	$[g/s]$
$q_{2,1}$	0	$[g/s]$
$q_{2,2}$	43.517	$[l/s]$
$q_{3,1}[p_{sys}]_1$	0	$[mol/s]$
$q_{3,1}[p_{sys}]_2$	0	$[mol/s]$
$q_{3,1}[p_{sys}]_3$	0	$[mol/s]$
$q_{3,1}[p_{sys}]_4$	0	$[MJ/s]$
$q_{3,2}[p_{sys}]_1$	10	$[mol/s]$
$q_{3,2}[p_{sys}]_2$	0	$[mol/s]$
$q_{3,2}[p_{sys}]_3$	10	$[mol/s]$
$q_{3,2}[p_{sys}]_4$	1.2	$[MJ/s]$

Proof. The proof of this proposition is given in Appendix B. □

Some useful consequences of Proposition 3.2.1 can be stated as follows: (1) If all reactors are non-idle, then $\alpha_{J_{nid}}(\cdot) = \alpha_{J_{tot}}(\cdot)$ is continuous. (2) If there are idle reactors and condition (3.34) is not satisfied for an evaluation point, $\alpha_{J_{nid}}(\cdot)$ is locally discontinuous at this point. We illustrate this through the following example.

Example (continued)**.** *Let us select a steady-state P^* of the 2-rector network (3.23), in which reactor 1 is idle and reactor 2 is non-idle. We set $c_{A1} = c_{B1} = 0$ and $c_{C1} = 1/22.4$ [mol/l] to represent the case where reactor 1 contains only component C and thus there are no reactions taking place. The steady state P^* is shown in Table 3.4.*

Using Eq. (3.26), we can evaluate J_{id} at P^* as

$$J_{id}|_{P^*} = \begin{pmatrix} -0.002 & 0 & 0 & 0 \\ 0.0014 & -0.0001 & 0 & 0 \\ -0.0020 & -0.0001 & 0 & 0 \\ 17.6416 & 0.1789 & 0 & 0 \end{pmatrix}, \tag{3.35}$$

with

$$\alpha_{J_{id}}|_{P^*} = 0. \tag{3.36}$$

https://doi.org/10.51202/9783186952035 - Generiert durch IP 216.73.216.36, am 18.01.2026, 21:11:22. © Urheberrechtlich geschützter Inhalt. Ohne gesonderte Erlaubnis ist jede urheberrechtliche Nutzung untersagt, insbesondere die Nutzung des Inhalts im Zusammenhang mit, für oder in KI-Systemen, KI-Modellen oder Generativen Sprachmodellen.

Actually, J_{id} at P^* has two zero eigenvalues. One corresponds to concentration c_{C1} and the other to temperature T_1. Note that, any other steady state of the idle reactor also results in multiple zero eigenvalues.

Using Eq. (3.27), we can evaluate J_{nid} at P^* as

$$J_{nid}|_{P^*} = \begin{pmatrix} -0.4452 & 0 & -0.0100 & -0.0001 \\ 0.0068 & -0.4355 & 0.0068 & 0.0001 \\ -0.0100 & -0.0003 & -0.4452 & -0.0001 \\ 88.7608 & 0.9002 & 88.7759 & 0.0281 \end{pmatrix},$$

with

$$\alpha_{J_{nid}}|_{P^*} = -0.0010. \tag{3.37}$$

Obviously, the operating point P^* for reactor 2 is stable.

Now we illustrate the discontinuity of $\alpha_{J_{nid}}(\cdot)$ at P^* (refer also to the proof in Appendix B). To show the discontinuity, we activate the idle reactor 1 by increasing the flowrate variable $q_{1,1}$ to a small positive number, i.e., $q_{1,1} = \epsilon > 0$. We use P' to denote this new operating point. At P', both reactors 1 and 2 are non-idle. So

$$J_{nid}|_{P'} = J_{tot} \approx \begin{pmatrix} J_{id}|_{P^*} & 0 \\ 0 & J_{nid}|_{P^*} \end{pmatrix}. \tag{3.38}$$

"$\approx$" holds, because all elements in $J_{tot}(\cdot)$ are continuous functions. We see that, at P^* J_{nid} is a 4×4 matrix, while at P' J_{nid} is a 8×8 matrix. So the size of J_{nid} is dependent on its arguments. From Eq. (3.38), we have

$$\alpha_{J_{nid}}|_{P'} \approx max\{\alpha_{J_{id}}|_{P^*}, \alpha_{J_{nid}}|_{P^*}\} = 0. \tag{3.39}$$

Denote $\epsilon_k > 0$, $k = 1, 2, \cdots$, as a sequence, which approaches 0. P'_k, $k = 1, 2, \cdots$, are evaluation points with respect to $q_{1,1} = \epsilon_k$. Because Eq. (3.39) always holds for any P'_k,

$$\lim_{k\to\infty} \alpha_{J_{nid}}|_{P'_k} = 0.$$

Combining with Eq. (3.37), we see that for the sequence $P'_k \to P^*$, sequence $\alpha_{J_{nid}}|_{P'_k}$ does not converge to $\alpha_{J_{nid}}|_{P^*}$. So $\alpha_{J_{nid}}(\cdot)$ is discontinuous at P^*.

An important conclusion is that, if there are idle reactors in a final optimal design, the eigenvalue constraint (3.33) is always at a discontinuous point. So, if we directly use Eq. (3.33) as a constraint of an optimal design problem, this final design can not be found by smooth optimization methods. ☐

3.2.3 A direct problem formulation

In this subsection, we first propose a direct problem formulation by using the eigenvalue constraint (3.33) for open-loop reactor network synthesis. Because of the difficulties to treat the discontinuity of constraint (3.33), integer variables are introduced in the next subsection to transform the direct problem formulation into an equivalent mixed-integer optimization problem. A tailored two-step solution approach will be presented for the transformed mixed-integer problem in Section 5.6.

https://doi.org/10.51202/9783186952035 - Generiert durch IP 216.73.216.36, am 18.01.2026, 21:11:22. © Urheberrechtlich geschützter Inhalt. Ohne gesonderte
Erlaubnis ist jede urheberrechtliche Nutzung untersagt, insbesondere die Nutzung des Inhalts im Zusammenhang mit, für oder in KI-Systemen, KI-Modellen oder Generativen Sprachmodellen.

After introducing the stability constraint (3.33), we are ready to present the novel problem formulation for open-loop reactor network synthesis as follows:

$$\begin{aligned}
&min_{x,q,p,p_{sys}} \varphi(x,q,p,p_{sys}) && (3.40a)\\
s.t.\ &0 = f_i(x_i,\cdots,q_{h(i,k)}g_{h(i,k)}(x_{\bar{h}(i,k)},p_i),\cdots, &&\\
&\quad q_{N+1,i}p_{sys},q_{i,1},\cdots,q_{i,N},p_i), \forall i=1,\cdots,N, && (3.40b)\\
&-c \geq \alpha_{J_{nid}}(x,q,p,p_{sys}), \forall \pi_\tau \in [\bar{\pi}_\tau - \Delta\bar{\pi}_\tau, \bar{\pi}_\tau + \Delta\bar{\pi}_\tau], && (3.40c)\\
&0 \geq h(x,q,p,p_{sys}). && (3.40d)
\end{aligned}$$

Eq. (3.40a), with the economic cost function $\varphi(\cdot)$, is the objective of the design optimization. Eq. (3.40b) refers to a steady-state of the model. Eq. (3.40c) is the so-called robust eigenvalue constraint. $\pi_\tau \in \mathbb{R}^{n_{\pi_\tau}}$ denotes a vector of uncertain parameters concatenating some specific elements from the vector of design parameters $(q^T, p^T, p_{sys}^T)^T$. The nominal values are $\bar{\pi}_\tau$ and the uncertainty is quantified by $\Delta\bar{\pi}_\tau$. $c > 0$ is a given constant. Eq. (3.40c) guarantees that the SA of the Jacobian of non-idle model is less than $-c$ for all realizations π_τ in the uncertain region $[\bar{\pi}_\tau \pm \Delta\bar{\pi}_\tau]$ and hence ensures robust dynamic properties of the final design. Eq. (3.40d) denotes other feasibility constraints on states and parameters, such as non-negative flowrate variables, upper bounds on the reactor volume, or ranges of temperature. Note that integer (binary) variables have not been introduced in problem (3.40).

The solution of problem (3.40) is very difficult because of two related reasons. As discussed in Section 3.1.6, the model is of variable structure because of the undetermined flowrate $q_{i,j}$, which results in a transition of reactors from non-idle to idle mode or vice versa. Accordingly, the eigenvalue spectrum of the non-idle model and hence the SA of non-idle subsystem change discontinuously. Even worse, if condition (3.34) is violated at an optimal solution, refer also to Eqs. (3.36) and (3.37) for the allyl chloride example, this solution will be exactly at a discontinuous point of Eq. (3.40c). Hence, any (standard) local NLP solver would most likely not solve the problem properly.

Therefore, we will reformulate problem (3.40) in the next subsection into a MINLP problem such that all of its constraints are continuous (smooth almost everywhere).

3.2.4 Problem reformulation

In this subsection, we present a mixed-integer reformulation of problem (3.40). The obtained MINLP is equivalent to the original one (in the sense of having the same optimal solution), but discontinuity in the eigenvalue constraint (3.40c) is replaced by discontinuities introduced by integer variables. The obtained MINLP can be better treated by smooth optimization methods, as detailed in Chapter 5.

We use integer variables $z_i \in \{0,1\}$, $i = 1,\cdots,N$, to indicate whether reactor i is non-idle ($z_i = 1$) or idle ($z_i = 0$). Denote $z = (z_1,\cdots,z_N)^T$. The disjunctions

$$\begin{bmatrix} z_i \\ \sum\limits_{j=1}^{N} q_{i,j} + \sum\limits_{k=1}^{N} q_{h(i,k)} > 0 \end{bmatrix} \vee \begin{bmatrix} \bar{z}_i \\ q_{i,j} = 0, \forall j = 1,\cdots,N \\ q_{h(i,k)} = 0, \forall k = 1,\cdots,N \end{bmatrix}, \forall i = 1,\cdots,N, \qquad (3.41)$$

represent idle or non-idle reactors. Because all $q_{i,j}$ are non-negative flowrate variables, in case of a non-idle reactor ($z_i = 1$) at least one $q_{i,j}$ or one $q_{h(i,k)}$ is positive. When all $q_{i,j} = 0$ and all $q_{h(i,k)} = 0$, the reactor is idle according to Definition 3.1.1.

https://doi.org/10.51202/9783186952035 - Generiert durch IP 216.73.216.36, am 18.01.2026, 21:11:22. © Urheberrechtlich geschützter Inhalt. Ohne gesonderte Erlaubnis ist jede urheberrechtliche Nutzung untersagt, insbesondere die Nutzung des Inhalts im Zusammenhang mit, für oder in KI-Systemen, KI-Modellen oder Generativen Sprachmodellen.

We introduce a matrix $\bar{J}$ defined by

$$\bar{J}(x,q,p,p_{sys},z) := J_{tot}(x,q,p,p_{sys}) - M \cdot \begin{bmatrix} (1-z_1)I_1 & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & (1-z_N)I_N \end{bmatrix}. \tag{3.42}$$

$\bar{J}$ has the same dimension as $J_{tot} \in \mathbb{R}^{n_x \times n_x}$. $I_i \in \mathbb{R}^{n_{x_i} \times n_{x_i}}$, $i = 1, \cdots, N$, is an identity matrix. n_{x_i} refers to the dimension of the state variables of x_i and $n_x = \sum_i n_{x_i}$. M denotes a sufficiently large positive constant.

According to the following proposition we can replace J_{nid} in problem (3.40) by $\bar{J}$ to obtain a continuous but still non-smooth eigenvalue constraint.

Proposition 3.2.2. *Assume that the elements in $J_{tot}(x,q,p,p_{sys})$ are bounded. For sufficiently large M, if Eq. (3.41) holds and if $I_{nid}(q) \neq \emptyset$, we have*

$$\alpha_{J_{nid}}(x,q,p,p_{sys}) = \alpha_{\bar{J}}(x,q,p,p_{sys},z). \tag{3.43}$$

Proof. The proof of this proposition is shown in Appendix C. Note that condition $\mathcal{I}_{nid}(q) \neq \emptyset$ is necessary, because J_{nid} is not defined for a reactor network with no non-idle reactors. □

The left hand side of Eq. (3.43) is a discontinuous function of continuous variables, while the right hand side of Eq. (3.43) is a continuous function of continuous and integer variables. If we replace constraint (3.40c) in problem (3.40) by Eq. (3.43) and use Eq. (3.41) as constraints, we obtain the reformulated problem

$$min_{x,q,p,p_{sys},z} \varphi(x,q,p,p_{sys}) \tag{3.44a}$$

$$s.t.\ 0 = f_i(x_i, \cdots, q_{h(i,k)} g_{h(i,k)}(x_{\bar{h}(i,k)}, p_i), \cdots,$$

$$q_{h(N+1,i)} p_{sys}, q_{i,1}, \cdots, q_{i,N}, p_i), \forall i = 1, \cdots, N, \tag{3.44b}$$

$$-c \geq \alpha_{\bar{J}}(x,q,p,p_{sys},z), \forall \pi_\tau \in [\bar{\pi}_\tau - \Delta\bar{\pi}_\tau, \bar{\pi}_\tau + \Delta\bar{\pi}_\tau], \tag{3.44c}$$

$$\begin{bmatrix} z_i \\ \sum\limits_{j=1}^{N} q_{i,j} + \sum\limits_{k=1}^{N} q_{h(i,k)} > 0 \end{bmatrix} \vee \begin{bmatrix} \bar{z}_i \\ q_{i,j} = 0, \forall j = 1, \cdots, N \\ q_{h(i,k)} = 0, \forall k = 1, \cdots, N \end{bmatrix}, \forall i = 1, \cdots, N, \tag{3.44d}$$

$$0 \geq h(x,q,p,p_{sys}), \tag{3.44e}$$

$$z_i \in \{0,1\}, \forall i = 1, \cdots, N. \tag{3.44f}$$

In problem (3.44) the reformulated constraint (3.44c) guarantees robust dynamic properties of non-idle reactors. $\alpha_{\bar{J}}(\cdot)$ is a standard eigenvalue function, which is continuous (but generally non-smooth) with respect to its arguments. Therefore, the methods to treat standard eigenvalue constraints can applied, refer to Section 5.4. Problem (3.44) will be solved by a two-step solution method proposed in Section 5.6.

A favorable feature of problem (3.44) is that the number of integer variables equals the number of reactors in the superstructure. Because there are not too many reactors in a typical reactor network, say less than 10, problem (3.44) contains typically only a few integers. This makes the problem easier to solved.

https://doi.org/10.51202/9783186952035 - Generiert durch IP 216.73.216.36, am 18.01.2026, 21:11:22. © Urheberrechtlich geschützter Inhalt. Ohne gesonderte
Erlaubnis ist jede urheberrechtliche Nutzung untersagt, insbesondere die Nutzung des Inhalts im Zusammenhang mit, für oder in KI-Systemen, KI-Modellen oder Generativen Sprachmodellen.

3.3 Summary

In this section, we have formulated an optimization problem (3.44) for the open-loop reactor network synthesis problem with guaranteed dynamic performance. The basic problem settings are adopted from [87], including the reactor network superstructure shown in Fig. 3.1. However, this work differs from and extends [87] in the following aspects.

First, idle reactors are allowed to appear in the superstructure. Designers can therefore decide on the optimal number of non-idle reactors in the final design. Eq. (3.33) is derived from Eq. (2.17), but they are not exactly the same. Eq. (3.33) measures the dynamic properties of only non-idle reactors, allowing the treatment of the dynamic properties of non-idle and idle reactors separately.

Second, parametric uncertainty is considered in the problem formulation, resulting in a robust design. Not only the nominal operating point is required to have the specified dynamic properties, but also the nearby operating points in the uncertainty region.

Third, in contrast to the conservative bounding method applied in [87], more advanced methods for treating eigenvalue constraints are employed in this work (cf. Section 5.4 and Section 5.6). This results in a more accurate treatment of the stability constraint and less conservative computational results. Fourth, the proposed formulation carries over to reactor network superstructures with both CSTR and PFR (cf. Appendix A), which allows decision making on the used type of reactors.

The current approach for the open-loop reactor network synthesis problem is subject to the following limitations. First, the eigenvalue constraint (3.44c) may get ill-conditioned after applying the big-M reformulation (3.42), if a too large M is chosen. To avoid this problem, a tight estimation of the smallest M is necessary. Second, there may exist other alternative reformulation strategies, rather than the proposed big-M reformulation (3.43), to treat the discontinuity problem of the formulated eigenvalue constraint (3.40c). Numerical performance should be evaluated and compared for various possible reformulations. Third, problem (3.44) is a MINLP problem with a non-smooth robust eigenvalue constraint and disjunctions, which is very challenging to solve, even if only local minima are of interest (cf. Section 5.6). One should therefore consider improving the solution procedure for the derived optimization problem.

https://doi.org/10.51202/9783186952035 - Generiert durch IP 216.73.216.36, am 18.01.2026, 21:11:22. © Urheberrechtlich geschützter Inhalt. Ohne gesonderte Erlaubnis ist jede urheberrechtliche Nutzung untersagt, insbesondere die Nutzung des Inhalts im Zusammenhang mit, für oder in KI-Systemen, KI-Modellen oder Generativen Sprachmodellen.

4 Simultaneous design of reactor network and its decentralized control system

The goal of simultaneous process and control design of reactor networks is to find the optimal flowsheet structure, operating conditions, process design parameters, as well as the control structure and control design parameters in an integrated step. The advantages of using a simultaneous design approach have been discussed in Section 1.3. Major tasks of simultaneous design of a reactor network and its control system are conceptually illustrated in Fig. 1.4, in which both flowsheet and control structure alternatives of a reactor network superstructure are degrees of freedom of the design problem.

This chapter is organized in a modeling and a problem formulation section. In the modeling section, we use complementarity constraints and disjunctions to formulate the selection of a decentralized control structure and its interaction with the selection of the flowsheet structure for closed-loop reactor network synthesis. In the problem formulation section, we first propose a robust eigenvalue constraint to ensure desired dynamic properties of the closed-loop reactor network. Then we use this constraint to formulate a semi-infinite MINLP for determining the optimal network design. This MINLP addresses the problem of simultaneous process and control design for closed-loop reactor networks with guaranteed robust dynamic properties. Solution methods for this optimization problem will be treated in Section 5.6.

4.1 Modeling of reactor networks with decentralized control structure

In this section, we present a closed-loop model for the simultaneous design of a reactor network and its decentralized PI control system. The closed-loop model is obtained by straightforwardly coupling multiple PI controllers to the open-loop reactor network model (3.20). Two sets of constraints, referring to control structure selection and to structural relations between reactors and controllers, will be introduced. Both flowsheet and control structure alternatives will be considered simultaneously, resulting in decision making on the process flowsheet and the control system structure.

4.1.1 A closed-loop reactor network model

We consider the open-loop model (3.20) to set up feedback control loops. Denote u_i and d_i as elements of vector p_i, which refer to manipulated variables of reactor i in addition to the flowrate variables and equipment design parameters of reactor i, respectively. The

https://doi.org/10.51202/9783186952035 - Generiert durch IP 216.73.216.36, am 18.01.2026, 21:11:22. © Urheberrechtlich geschützter Inhalt. Ohne gesonderte
Erlaubnis ist jede urheberrechtliche Nutzung untersagt, insbesondere die Nutzung des Inhalts im Zusammenhang mit, für oder in KI-Systemen, KI-Modellen oder Generativen Sprachmodellen.

open-loop model (3.20) can be rewritten as

$$\begin{aligned}\dot{x}_i = f_i(x_i, &\underbrace{\cdots, q_{h(i,k)} g_{h(i,k)}(x_{\bar{h}(i,k)}, u_{\bar{h}(i,k)}, d_{\bar{h}(i,k)}), \cdots}_{k=1,\cdots,N-1}, q_{h(i,N)} p_{sys},\\ &q_{i,1}, \cdots, q_{i,N}, u_i, d_i), \forall i = 1, \cdots, N.\end{aligned} \tag{4.1}$$

Now we extend model (4.1) by feedback control loops. We assume that all flowrate variables $q = (q_{1,1}, \cdots, q_{N,N}, q_{N+1,1}, \cdots, q_{N+1,N})^T \in \mathbb{R}^{(N+1)N}$, all u_i, $i = 1, \cdots, N$, and the energy density in the feed to the reactor network corresponding to the last element of p_{sys} can be manipulated. Therefore, we denote all the candidate MV of the reactor network by

$$u = (q^T, u_1^T, \cdots, u_N^T, [p_{sys}]_{N_c+1})^T \in \mathbb{R}^{n_m}. \tag{4.2}$$

Denote π as the vector

$$\pi = ([p_{sys}]_1, \cdots, [p_{sys}]_{N_c}, d_1^T, \cdots, d_N^T)^T, \tag{4.3}$$

which contains all equipment and process design parameters, which can not be manipulated. $[p_{sys}]_l$, $l = 1, \cdots, N_c + 1$, denotes the l-th element of vector p_{sys}.

Denote $[u]_v$ as the v-th element in vector u, $v = 1, \cdots, n_m$. The following definition relates (one or more) individual candidate MV to an individual reactor.

Definition 4.1.1 (Candidate MV of reactor i). *A candidate MV, $[u]_v$, $v \in \{1, \cdots, n_m\}$, is called a candidate MV of reactor i, if $[u]_v$ is an element of vector u_i or vector $(q_{i,1}, \cdots, q_{i,N}, q_{h(i,1)}, \cdots, q_{h(i,N)})^T$.*

We note that, although u_i are candidate MV of a single reactor i, elements of $(q_{i,1}, \cdots, q_{i,N}, q_{h(i,1)}, \cdots, q_{h(i,N)})^T$, except for $q_{h(i,N)}$ (corresponding to the inlet connection with the splitter) and $q_{i,N}$ (corresponding to the outlet connection with the mixer), are candidate MV of two different reactors. Due to interconnections, $q_{h(i,k)}$ is a candidate MV of both, reactors i and $\bar{h}(i,k)$. Similarly, $q_{i,j}$ is a candidate MV of both, reactors i and $\bar{l}(i,j)$.

We define index sets Θ_i for reactor i, $i = 1, \cdots, N$, so that if $v^* \in \Theta_i$, $[u]_{v^*}$ is a candidate MV of reactor i according to Definition 4.1.1. For $i = 1, \cdots, N$, we introduce

$$\Theta_i := \{v \in \{1, \cdots, n_m\} \mid [u]_v \text{ is a candidate MV of reactor } i. \}.$$

Obviously, because each $q_{i,j}$ belongs to the candidate MV of two reactors, $\Theta_{i_1} \cap \Theta_{i_2} \neq \emptyset$, if $i_1 \neq i_2$.

Denote $y_{i,r} \in \mathbb{R}$, $i = 1, \cdots, N$, $r = 1, \cdots, n_c^i$, as a candidate control variable (CV). $y_{i,r}$ refers to the r-th candidate CV of reactor i, which is physically measured. n_c^i is the total number of candidate CV of reactor i. Consider that

$$y_{i,r} = \phi_{i,r}(x_i, d_i), i = 1, \cdots, N, r = 1, \cdots, n_c^i, \tag{4.4}$$

where $\phi_{i,r}(\cdot)$ is a scalar-valued smooth function. Denote $y = (y_{1,1}, \cdots, y_{N,n_c^N})^T \in \mathbb{R}^{n_c}$ as a vector containing all candidate CV $y_{i,r}$ for all reactors. $n_c = \sum_i n_c^i$ refers to the total number of candidate CV of the network. We use the symbol w, $w = 1, \cdots, n_c$, to indicate the w-th element $[y]_w$ in vector y. This index w can be related to the subindex (i, r) of $y_{i,r}$ by introducing a function $\varrho(i, r)$, so that $y_{i,r}$ is the $\varrho(i, r)$-th element in vector y, i.e.,

$$y_{i,r} = [y]_{\varrho(i,r)}, i = 1, \cdots, N, r = 1, \cdots, n_c^i. \tag{4.5}$$

https://doi.org/10.51202/9783186952035 - Generiert durch IP 216.73.216.36, am 18.01.2026, 21:11:22. © Urheberrechtlich geschützter Inhalt. Ohne gesonderte
Erlaubnis ist jede urheberrechtliche Nutzung untersagt, insbesondere die Nutzung des Inhalts im Zusammenhang mit, für oder in KI-Systemen, KI-Modellen oder Generativen Sprachmodellen.

We also use $\varrho^{-1}(\cdot)$ to denote the inverse function of $\varrho(\cdot)$.

Any (not necessarily decentralized) feedback control system involving PI controllers can be modeled by

$$\dot{e} = y - \bar{y}, \tag{4.6a}$$

$$u = \bar{u} + K(y - \bar{y} + Te), \tag{4.6b}$$

where $e_{i,r} \in \mathbb{R}$ denotes the state variable of the (i,r)-th controller [124]. $e := (e_{1,1}, \cdots, e_{N,n_c^N})^T \in \mathbb{R}^{n_c}$ collects the states of all n_c candidate PI controllers. $K \in \mathbb{R}^{n_m \times n_c}$ is the proportional control gain matrix, which is in general non-square. Index v, $v = 1, \cdots, n_m$, and index w, $w = 1, \cdots, n_c$, are used to indicate the rows and columns of matrix K, respectively. $T := diag(1/t_{1,1}, \cdots, 1/t_{N,n_c^N}) \in \mathbb{R}^{n_c \times n_c}$ is the integral control gain matrix, which is square and diagonal. $t_{i,r} \in \mathbb{R}$ refers to the integral control gain of the (i,r)-th controllers. $\bar{u} \in \mathbb{R}^{n_m}$ denotes a vector of offset values of u and $\bar{y} \in \mathbb{R}^{n_c}$ denotes a vector of reference signals of y. Analogously, the offset values of the flow rates q (elements of $\bar{u}$) are denoted by $\bar{q}$. Note that the control structure embedded in Eq. (4.6) is not fixed. It is rather determined by the zero/non-zero patterns of matrix K.

Combining Eqs. (4.1) and (4.6) and eliminating y through Eq. (4.4), a closed-loop model of a reactor network with a flexible flowsheet and control structure can be formulated as

$$\begin{aligned} \dot{x}_i = f_i(x_i, & \underbrace{\cdots, q_{h(i,k)} g_{h(i,k)}(x_{\bar{h}(i,k)}, u_{\bar{h}(i,k)}, d_{\bar{h}(i,k)}), \cdots}_{k=1,\cdots,N-1}, q_{h(i,N)} p_{sys}, \\ & q_{i,1}, \cdots, q_{i,N}, u_i, d_i), \;\; i = 1, \cdots, N, \end{aligned} \tag{4.7a}$$

$$\dot{e}_{i,r} = \phi_{i,r}(x_i, d_i) - \bar{y}_{i,r}, \;\; i = 1, \cdots, N, \; r = 1, \cdots, n_c^i, \tag{4.7b}$$

$$[u]_v = [\bar{u}]_v + \sum_{\forall (i,r)} [K]_{v,\varrho(i,r)} (\phi_{i,r}(x_i, d_i) - \bar{y}_{i,r} + \frac{1}{t_{i,r}} e_{i,r}), \;\; v = 1, \cdots, n_m, \tag{4.7c}$$

where $[K]_{v,\varrho(i,r)}$ denotes the $(v, \varrho(i,r))$-th element in matrix K.

We denote $K_v = ([K]_{1,1}, \cdots, [K]_{n_m,n_c})^T \in \mathbb{R}^{n_m \cdot n_c}$ as a vector concatenating all variables in matrix K and $T_v := (t_{1,1}, \cdots, t_{N,n_c^N})^T \in \mathbb{R}^{n_c}$ a vector concatenating all variables in matrix T. Eq. (4.7) contains $n_x + n_c + n_m$ equations, referring to the dimensions of variables $x \in \mathbb{R}^{n_x}$, $e \in \mathbb{R}^{n_c}$ and $u \in \mathbb{R}^{n_m}$. The degrees of freedom of the closed-loop model (4.7) are therefore

$$\psi_c = (\pi^T, \bar{u}^T, \bar{y}^T, K_v^T, T_v^T)^T. \tag{4.8}$$

System (4.7) models a closed-loop reactor network with flexible flowsheet and control structures. By determining the degrees of freedom ψ_c, model (4.7) can realize different flowsheet and control structures. As it will be shown later, the offset values $\bar{q}$ (elements in vector $\bar{u}$), which represent the nominal flowrates of flow connections, will be used to determine the flowsheet structure, while variables K_v (variables in matrix K) will be used to determine the structure of the decentralized control system. In the following subsections, we will impose additional constraints, such that a decentralized control structure and feasible structural relationships between reactors and controllers can be ensured in the final design.

https://doi.org/10.51202/9783186952035 - Generiert durch IP 216.73.216.36, am 18.01.2026, 21:11:22. © Urheberrechtlich geschützter Inhalt. Ohne gesonderte Erlaubnis ist jede urheberrechtliche Nutzung untersagt, insbesondere die Nutzung des Inhalts im Zusammenhang mit, für oder in KI-Systemen, KI-Modellen oder Generativen Sprachmodellen.

4.1.2 Complementarity constraints for control structure selection

By setting the elements of matrix K to zero or non-zero values, Eq. (4.6) can realize different degrees of centralization or decentralization of the control system. For example, a fully centralized control structure can be obtained by setting a dense matrix K, while a partially decentralized control structure can be obtained by requiring K to comprise of several non-zero and zero submatrices. In this subsection, we propose complementarity-based design constraints which ensure a fully decentralized control structure.

In a fully decentralized control structure each single candidate MV (an element in the vector u) is paired with a single candidate CV (an element in the vector y) and vice versa. Each row and each column of matrix K has to contain at most one non-zero element [124]. Hence, the task of designing a decentralized control structure can be transformed into the task of determining the locations of zero and non-zero elements in matrix K.

The following lemma is elementary and is straightforward to prove.

Lemma 4.1.1. *Consider any vector* $\xi \in \mathbb{R}^{n_\xi}$, $\xi \geq 0$. *Denote* $[\xi]_i$ *as the i-th element of* ξ, $i \in \{1, \cdots, n_\xi\}$. *If the complementarity constraints*

$$0 \leq [\xi]_i \perp \sum_{i' \neq i} [\xi]_{i'} \geq 0, \ \forall i = 1, \cdots, n_\xi, \tag{4.9}$$

hold, then either $\xi = 0$ *or there is at most one positive element in vector* ξ.

Proof. If $\xi = 0$, then Eq. (4.9) holds. If $[\xi]_{i^*} > 0$ and $[\xi]_{i'} = 0$, $\forall i' \neq i^*$, i.e. exactly one element of ξ is positive, then Eq. (4.9) holds. Assume now that $[\xi]_{i_1} > 0$ and $[\xi]_{i_2} > 0$, $i_1 \neq i_2$, i.e., at least two elements of ξ are positive. Because $\xi \geq 0$, we have

$$\begin{aligned} \sum_{i' \neq i_1} [\xi]_{i'} \geq \xi_{i_2} > 0, \\ \sum_{i' \neq i_2} [\xi]_{i'} \geq \xi_{i_1} > 0. \end{aligned} \tag{4.10}$$

Therefore Eq. (4.9) is violated for both $i = i_1$ and $i = i_2$. □

Denote $\hat{K}$, K^+, $K^- \in \mathbb{R}^{n_m \times n_c}$ as three matrices with the size of K. $[\hat{K}]_{v,w}$, $[K^+]_{v,w}$ and $[K^-]_{v,w}$ denote their (v, w)-th element. The proposed complementarity constraints are

$$[K]_{v,w} = [K^+]_{v,w} - [K^-]_{v,w}, \quad v = 1, \cdots, n_m, w = 1, \cdots, n_c, \tag{4.11a}$$

$$[\hat{K}]_{v,w} = [K^+]_{v,w} + [K^-]_{v,w}, \quad v = 1, \cdots, n_m, w = 1, \cdots, n_c, \tag{4.11b}$$

$$0 \leq [K^+]_{v,w} \perp [K^-]_{v,w} \geq 0, \quad v = 1, \cdots, n_m, w = 1, \cdots, n_c, \tag{4.11c}$$

$$0 \leq [\hat{K}]_{v,w} \perp \sum_{w' \neq w} [\hat{K}]_{v,w'} \geq 0, \quad v = 1, \cdots, n_m, w = 1, \cdots, n_c, \tag{4.11d}$$

$$0 \leq [\hat{K}]_{v,w} \perp \sum_{v' \neq v} [\hat{K}]_{v',w} \geq 0, \quad v = 1, \cdots, n_m, w = 1, \cdots, n_c. \tag{4.11e}$$

Eqs. (4.11a)-(4.11c) ensure that K^+ contains only the positive elements of K, K^- contains only the absolute values of the negative elements of K, and $\hat{K}$ contains the absolute values of all elements of K. In particular, Eq. (4.11c) guarantees that both $[K^+]_{v,w}$ and $[K^-]_{v,w}$ are non-negative, and at least one of them is zero. Eq. (4.11a) therefore guarantees that (i) if $[K]_{v,w} \geq 0$, then $[K^+]_{v,w} = [K]_{v,w} \geq 0$, $[K^-]_{v,w} = 0$; (ii) if $[K]_{v,w} \leq 0$, then $[K^+]_{v,w} = 0$, $[K^-]_{v,w} = -[K]_{v,w} \geq 0$. Hence, $[\hat{K}]_{v,w}$ is equal to the absolute value of $[K]_{v,w}$.

https://doi.org/10.51202/9783186952035 - Generiert durch IP 216.73.216.36, am 18.01.2026, 21:11:22. © Urheberrechtlich geschützter Inhalt. Ohne gesonderte Erlaubnis ist jede urheberrechtliche Nutzung untersagt, insbesondere die Nutzung des Inhalts im Zusammenhang mit, für oder in KI-Systemen, KI-Modellen oder Generativen Sprachmodellen.

Since all elements in $\hat{K}$ are non-negative, Eq. (4.9) applied to each rows and columns of $\hat{K}$ results in Eqs. (4.11d)-(4.11e). Eq. (4.11d) guarantees that each row of $\hat{K}$ has at most one positive element. Analogous arguments can be applied to Eq. (4.11e), which guarantees that each column of $\hat{K}$ has at most one positive element. Since $\hat{K}$ contains the absolute values of the elements of K, Eqs. (4.11d), (4.11e) guarantee that each row and each column of K has at most one non-zero element.

Note that, Eq. (4.11) allows zero rows in matrix K. If the v^*-th row of K contains only zeros, i.e., if

$$[K]_{v^*,w} = 0, \quad \forall w = 1, \cdots, n_c, \tag{4.12}$$

$[u]_{v^*}$ is not subject to any control (cf. Eq. (4.7c), $[u]_{v^*} = [\bar{u}]_{v^*}$). Eq. (4.11) also allows that certain columns of matrix K contain only zero values. If this is true, for the w^*-th column, i.e., if

$$[K]_{v,w^*} = 0, \quad \forall v = 1, \cdots, n_m,$$

then $[y]_{w^*}$ is an unmeasured variable and it is not used to form any closed loop[1]. In the extreme case, if $K = \mathbf{0}$ and therefore $K^+ = K^- = \hat{K} = \mathbf{0}$, the open-loop case is recovered.

Eq. (4.11) is equivalent to the integer-based formulation proposed in [124] for control structure design. The authors formulate the same rules for pairing candidate MV and CV, i.e., a single candidate MV is paired with at most one candidate CV, and vice versa. However, in contrast to [124], Eq. (4.11) does not use any integer variables. Complementarity constraints allow a more efficient numerical treatment, including smoothing methods [16, 56].

4.1.3 Idle reactors and controllers

The closed-loop reactor network model (4.7) contains N candidate reactors and n_c candidate PI controllers. Idle reactors and controllers refer to the reactors and controllers in the final design, which will not be physically realized. Therefore, one needs to figure out how many and which reactors and controllers should be used in the final design. In this subsection, we first adapt the definitions of idle reactors introduced in Section 3.1.6 to the case of closed-loop reactor network design and then formalize the definition of idle controllers.

Definition 4.1.2 (Idle reactors in a closed-loop reactor network)**.** *A reactor i, $i = 1, \cdots, N$, is idle, if*

$$\begin{aligned} \bar{q}_{i,j} &= 0, \forall j = 1, \cdots, N, \\ \bar{q}_{h(i,k)} &= 0, \forall k = 1, \cdots, N. \end{aligned} \tag{4.13}$$

Otherwise, reactor i is called non-idle.

$\bar{q}_{i,j}$ denotes the offset value of flowrate variable $q_{i,j}$ (a candidate MV), which refers to the j-th outlet of reactor i. $\bar{q}_{h(i,k)}$ denotes the offset value of flowrate variable $q_{h(i,k)}$ (a candidate MV), which refers to k-th inlet of reactor i. Eq. (4.13) requires that the offset values of all inlet and outlet flowrates of idle reactor i are zero, which represents the fact that reactor i has no inlets and outlets. Hence, fixing the real, non-negative values of $\bar{q}$ to zero or a positive quantity determines the flowsheet structure without the need for integers as in the established synthesis approaches.

---

[1] In this case, the $\varrho^{-1}(w^*)$-th controller will not be physically realized, refer also to Definition 4.1.3.

https://doi.org/10.51202/9783186952035 - Generiert durch IP 216.73.216.36, am 18.01.2026, 21:11:22. © Urheberrechtlich geschützter Inhalt. Ohne gesonderte Erlaubnis ist jede urheberrechtliche Nutzung untersagt, insbesondere die Nutzung des Inhalts im Zusammenhang mit, für oder in KI-Systemen, KI-Modellen oder Generativen Sprachmodellen.

We use (binary) integer variables $z_i \in \{0,1\}$, $i = 1, \cdots, N$, to indicate whether reactor i is idle ($z_i = 0$) or non-idle ($z_i = 1$). $z_r = (z_1, \cdots, z_N)^T$ encodes the existence or non-existence of each reactor. Definition 4.1.2 leads to the disjunctions

$$\begin{bmatrix} z_i \\ \sum\limits_{j=1}^{N} \bar{q}_{i,j} + \sum\limits_{k=1}^{N} \bar{q}_{h(i,k)} > 0 \end{bmatrix} \vee \begin{bmatrix} \bar{z}_i \\ \bar{q}_{i,j} = 0, \forall j = 1, \cdots, N \\ \bar{q}_{h(i,k)} = 0, \forall k = 1, \cdots, N \end{bmatrix}, \forall i = 1, \cdots, N. \tag{4.14}$$

Note that, because $\bar{q}$ are non-negative variables, Eq. (4.14) guarantees that, either at least one $\bar{q}_{i,j}$ or one $\bar{q}_{h(i,k)}$ is positive, or all of them are zero.

Accordingly, idle controllers are the ones which are not included in the final design:

Definition 4.1.3 (Idle controller)**.** *For $i \in \{1, \cdots, N\}$, $r \in \{1, \cdots, n_c^i\}$, the (i,r)-th PI controller is idle, if the $\varrho(i,r)$-th column of matrix K is zero:*

$$[K]_{v,\varrho(i,r)} = 0, \quad \forall v = 1, \cdots, n_m. \tag{4.15}$$

Otherwise, the (i,r)-th PI controller is called non-idle.

If the w^*-th column of matrix K is zero, a candidate CV $[y]_{w^*} = y_{\varrho^{-1}(w^*)}$ is not paired with any MV. This quantity does not need to be measured and the corresponding $\varrho^{-1}(w^*)$-th PI controller will not be included in the final design. Hence, we can define idle controllers by checking the columns of matrix K.

We use integer variables $z_{i,r} \in \{0,1\}$, $i = 1, \cdots, N$ and $r = 1, \cdots, n_c^i$, to indicate whether the (i,r)-th controller is idle ($z_{i,r} = 0$) or non-idle ($z_{i,r} = 1$). Denote $z_c = (z_{1,1}, \cdots, z_{N,n_c^N})^T$ to encode the existence of each PI controller. Definition 4.1.3 leads to the disjunctions

$$\begin{bmatrix} z_{i,r} \\ \sum\limits_{v=1}^{n_m} [\hat{K}]_{v,\varrho(i,r)} > 0 \end{bmatrix} \vee \begin{bmatrix} \bar{z}_{i,r} \\ \sum\limits_{v=1}^{n_m} [\hat{K}]_{v,\varrho(i,r)} = 0 \end{bmatrix}, \forall i = 1, \cdots, N, r = 1, \cdots, n_c^i. \tag{4.16}$$

Due to Eq. (4.11), the elements of $\hat{K}$ are equal to the absolute values of the elements of K. Eq. (4.16) therefore guarantees that either at least an element in the $\varrho(i,r)$-th column of matrix K is non-zero, or all elements in the $\varrho(i,r)$-th column of K are zero. For later discussion, denote $z = (z_r^T, z_c^T)^T = (z_1, \cdots, z_N, z_{1,1}, \cdots, z_{N,n_c^N})^T$ to collect all integer variables indicating the existence of each reactor and controller.

For convenience, we introduce the index sets

$$\begin{aligned}
\mathcal{I} &= \{i | i = 1, \cdots, N\}, \\
\mathcal{I}_{id}(\bar{q}) &= \{i \in \mathcal{I} \mid \text{reactor } i \text{ is idle}\}, \\
\mathcal{I}_{nid}(\bar{q}) &= \{i \in \mathcal{I} \mid \text{reactor } i \text{ is non-idle}\}, \\
\mathcal{U} &= \{v \mid v = 1, \cdots, n_m\}, \\
\mathcal{U}_{id}(K) &= \{v \in \mathcal{U} \mid [u]_v \text{ is not subject to control}\}, \\
\mathcal{U}_{nid}(K) &= \{v \in \mathcal{U} \mid [u]_v \text{ is subject to control}\}, \\
\mathcal{C} &= \{(i,r) \mid i = 1, \cdots, N, \ r = 1, \cdots, n_c^i\}, \\
\mathcal{C}_{id}(K) &= \{(i,r) \in \mathcal{C} \mid \text{controller } (i,r) \text{ is idle}\}, \\
\mathcal{C}_{nid}(K) &= \{(i,r) \in \mathcal{C} \mid \text{controller } (i,r) \text{ is non-idle}\}.
\end{aligned}$$

https://doi.org/10.51202/9783186952035 - Generiert durch IP 216.73.216.36, am 18.01.2026, 21:11:22. © Urheberrechtlich geschützter Inhalt. Ohne gesonderte Erlaubnis ist jede urheberrechtliche Nutzung untersagt, insbesondere die Nutzung des Inhalts im Zusammenhang mit, für oder in KI-Systemen, KI-Modellen oder Generativen Sprachmodellen.

For given sets $\mathcal{I}_{id}$, $\mathcal{I}_{nid}$, $\mathcal{C}_{id}$, $\mathcal{C}_{nid}$, we further define

$$\begin{aligned} x_{id} &:= (\cdots, x_i^T, \cdots)^T, i \in \mathcal{I}_{id}, \\ x_{nid} &:= (\cdots, x_i^T, \cdots)^T, i \in \mathcal{I}_{nid}, \\ e_{id} &:= (\cdots, e_{i,r}, \cdots)^T, (i,r) \in \mathcal{C}_{id}, \\ e_{nid} &:= (\cdots, e_{i,r}, \cdots)^T, (i,r) \in \mathcal{C}_{nid}. \end{aligned} \tag{4.17}$$

x_{id} and e_{id} refer to the states of idle reactors and the states of idle controllers, respectively. x_{nid} and e_{nid} refer to the states of non-idle reactors and the states of non-idle controllers, respectively.

4.1.4 Structural constraints

The constraints in the previous subsections encode different flowsheet and control structure alternatives as well as decisions regarding idle reactors and controllers and their interconnections by manipulating the flowrate offset variables $\bar{q}$ and the control gain matrix K. However, we are not allowed to choose $\bar{q}$ and K in an arbitrary manner to ensure feasible designs. For example, an inlet flowrate of an idle reactor should not be manipulated by a CV of a non-idle reactor, or the coolant flowrate of a non-idle reactor should not be manipulated by the temperature measurement of an idle reactor. Hence, we need additional design constraints, which ensure feasible and physically correct structural relationships among reactors and controllers as well as a proper pairing of candidate CV and MV.

We propose two kinds of structural relationships, which are required to hold in the final design:

1. All controllers belonging to idle reactors are idle:

   $$i \in \mathcal{I}_{id} \Rightarrow (i,r) \in \mathcal{C}_{id}, \;\; \forall i = 1, \cdots, N, \; \forall r = 1, \cdots, n_c^i. \tag{4.18}$$

   This relationship can be represented as

   $$z_{i,r} \leq z_i, \;\; r = 1, \cdots, n_c^i, \; i = 1, \cdots, N, \tag{4.19}$$

   with z_i and $z_{i,r}$ defined in Eqs. (4.14), (4.16) [140].

2. All candidate MV of idle reactors are not subject to control:

   $$v \in \bigcup_{i \in \mathcal{I}_{id}} \Theta_i \Rightarrow v \in \mathcal{U}_{id}. \tag{4.20}$$

   This relationship can be represented by the disjunction

   $$\begin{bmatrix} z_i \\ \emptyset \end{bmatrix} \vee \begin{bmatrix} \bar{z}_i \\ \sum\limits_{w=1}^{n_c} [\hat{K}]_{v,w} = 0, \forall v \in \Theta_i \end{bmatrix}, i = 1, \cdots, N. \tag{4.21}$$

Sometimes it is convenient to replace (4.18) by its contrapositive:

$$(i,r) \in \mathcal{C}_{nid} \Rightarrow i \in \mathcal{I}_{nid}, \;\; \forall i = 1, \cdots, N, \; \forall r = 1, \cdots, n_c^i. \tag{4.22}$$

However, we stress that the backward direction of (4.18) is not true, i.e.,

$$(i,r) \in \mathcal{C}_{id} \nRightarrow i \in \mathcal{I}_{id}, \tag{4.23}$$

because a non-idle reactor may have a measurement, which is not used to close a loop.

https://doi.org/10.51202/9783186952035 - Generiert durch IP 216.73.216.36, am 18.01.2026, 21:11:22. © Urheberrechtlich geschützter Inhalt. Ohne gesonderte
Erlaubnis ist jede urheberrechtliche Nutzung untersagt, insbesondere die Nutzung des Inhalts im Zusammenhang mit, für oder in KI-Systemen, KI-Modellen oder Generativen Sprachmodellen.

4.1.5 Structural properties of the closed-loop model

By imposing constraints (4.19) and (4.21) for the selection of $\bar{q}$ and K, the closed-loop model (4.7) has certain guaranteed structural properties, which are revealed in this subsection and allow for a decomposition of model (4.7).

Analysis of the formulations of candidate MV. We first analyze the formulation of the candidate MV concatenated in vector u and presented in Eq. (4.7c). Denote

$$u_{p1} := (\cdots, [u]_v, \cdots)^T,\ v \in \bigcup_{i \in \mathcal{I}_{id}} \Theta_i, \tag{4.24}$$

and u_{p2} as the other variables in u. After probably reordering the sequence, we have

$$u = (u_{p1}^T, u_{p2}^T)^T. \tag{4.25}$$

u_{p1} refers to the vector of candidate MV, which belong to at least one idle reactor. Relationship (4.20) or Eq. (4.21) guarantee that all elements in u_{p1} are not subject to control. Therefore, from Eq. (4.7c), we have

$$u_{p1} = \bar{u}_{p1}. \tag{4.26}$$

u_{p2} refers to the vector of candidate MV, which do not belong to the candidate MV of any idle reactor. Note that, elements in u_{p2} may or may not be subject to control, because (4.18) or (4.20) do not impose any conditions on the rows of matrix K corresponding to u_{p2}.

Consider $[u]_{v^*}$ to be an element of u_{p2}. Then

$$[u]_{v^*} = [\bar{u}]_{v^*} + \sum_{\forall (i,r) \in \mathcal{C}_{nid}} [K]_{v^*, \varrho(i,r)} (\phi_{i,r}(x_i, d_i) - \bar{y}_{i,r} + \frac{1}{t_{i,r}} e_{i,r}) \tag{4.27}$$

holds according to Eqs. (4.7c), (4.16). Obviously, u_{p2} only depends on x_{nid}, e_{nid} and ψ_c.

In summary, u_{p1} are always equal to their offset values $\bar{u}_{p1}$, and the candidate MV u_{p2} do not depend on x_{id} and e_{id}. u_{p1} refer to MV, which belong to at least one idle reactor. u_{p2} refer to MV of non-idle reactors, which may or may not be subject to control. We will use this property to analyze the structural relationships in the closed-loop model (4.7) later.

Submodels of the closed-loop reactor network. Next, we decompose the model (4.7) into submodels. For given values of $\bar{q}$ and K, we partition (4.7a), (4.7b) into state equations for idle reactors, non-idle reactors, idle controllers and non-idle controllers and eliminate u using Eq. (4.7c).

Consider that reactor i is idle. From Definition 4.1.1, $q_{i,j}$, $\forall j = 1, \cdots, N$, $q_{h(i,k)}$, $\forall k = 1, \cdots, N$, and u_i are the candidate MV of reactor i, and therefore they are elements of vector u_{p1} defined in Eq. (4.24). Using Eqs. (4.13), (4.26), the submodel of an idle reactor i can then be represented by

$$\begin{aligned} \dot{x}_i =& f_i(x_i, \cdots, \underbrace{q_{h(i,k)}}_{=0} g_{h(i,k)}(x_{\bar{h}(i,k)}, u_{\bar{h}(i,k)}, d_{\bar{h}(i,k)}), \cdots, \underbrace{q_{h(i,N)}}_{=0} p_{sys}, \underbrace{q_{i,1}}_{=0}, \cdots, \underbrace{q_{i,N}}_{=0}, \underbrace{u_i}_{=\bar{u}_i}, d_i) \\ :=& f_i(x_{id}, \psi_c), \quad i \in \mathcal{I}_{id}. \end{aligned} \tag{4.28}$$

https://doi.org/10.51202/9783186952035 - Generiert durch IP 216.73.216.36, am 18.01.2026, 21:11:22. © Urheberrechtlich geschützter Inhalt. Ohne gesonderte Erlaubnis ist jede urheberrechtliche Nutzung untersagt, insbesondere die Nutzung des Inhalts im Zusammenhang mit, für oder in KI-Systemen, KI-Modellen oder Generativen Sprachmodellen.

Note that this submodel depends neither on the states of non-idle reactors x_{nid} nor on the states of controllers e.

Likewise, the submodel of a non-idle reactor i becomes

$$\begin{aligned} \dot{x}_i = f_i(x_i, &\underbrace{\cdots, 0, \cdots}_{\substack{\text{inlets from} \\ \text{idle reactors}}}, \underbrace{\cdots, q_{h(i,k')} g_{h(i,k')}(x_{\bar{h}(i,k')}, u_{\bar{h}(i,k')}, d_{\bar{h}(i,k')}), \cdots}_{\substack{\text{inlets from} \\ \text{non-idle reactors}}}, \\ &\underbrace{q_{h(i,N)} p_{sys}}_{\substack{\text{inlets from} \\ \text{system feed}}}, \underbrace{\cdots, 0, \cdots}_{\substack{\text{outlets to} \\ \text{idle reactors}}}, \underbrace{\cdots, q_{i,j'}, \cdots}_{\substack{\text{outlets to} \\ \text{non-idle reactors}}}, \underbrace{q_{i,N}}_{\substack{\text{outlet to} \\ \text{system output}}}, u_i, d_i), \quad i \in \mathcal{I}_{nid}. \end{aligned} \tag{4.29}$$

In this equation, $q_{h(i,k')}$ and $q_{i,j'}$ refer to the flowrate of the k'-th inlet and the j'-th outlet of reactor i that are connected with other non-idle reactors. Therefore, $q_{h(i,k')}$ and $q_{i,j'}$ are not the candidate MV of any idle reactor and thus elements of vector u_{p2}. $q_{h(i,N)}$ refers to the feed of raw material and $q_{i,N}$ refers to the product outlet stream of reactor i. Because $q_{h(i,N)}$ and $q_{i,N}$ are the candidate MV of only reactor i (i.e. they are not candidate MV of other reactors, cf. Definition 4.1.1), $q_{h(i,N)}$ and $q_{i,N}$ are elements of vector u_{p2}. u_i and $u_{\bar{h}(i,k')}$ refer to the candidate MV of reactor i and non-idle reactor $\bar{h}(i,k')$, and therefore they are elements in u_{p2}. $x_{\bar{h}(i,k')}$ refers to the states of non-idle reactor $\bar{h}(i,k')$, while d_i and $d_{\bar{h}(i,k')}$ refer to design parameters. The first N_c elements of p_{sys} are design parameters, while the last element of p_{sys} is an element of u_{p2} according to Definition 4.1.1 and Eq. (4.24). Hence, all variables appearing on the right hand side of Eq. (4.29) are either state variables of non-idle reactors, elements of vector u_{p2} (u_{p2} depends only on x_{nid}, e_{nid} and ψ_c, cf. Eq. (4.27)), or elements of ψ_c. The submodels of non-idle reactors do not depend on the states of idle reactors x_{id} or the states of idle controllers e_{id}.

Accordingly, submodels of idle and non-idle controllers can be formulated as

$$\begin{aligned} \dot{e}_{i,r} =& \phi_{i,r}(x_i, d_i) - \bar{y}_{i,r} \\ :=& \phi_{i,r}(x_{id}, x_{nid}, \psi_c), \ (i,r) \in \mathcal{C}_{id}, \end{aligned} \tag{4.30}$$

and

$$\begin{aligned} \dot{e}_{i,r} =& \phi_{i,r}(x_i, d_i) - \bar{y}_{i,r} \\ :=& \phi_{i,r}(x_{nid}, \psi_c), \ (i,r) \in \mathcal{C}_{nid}, \end{aligned} \tag{4.31}$$

respectively. We stress that submodel (4.30) of idle controllers may depend on both x_{id} and x_{nid}, because x_i may still be a state variable of a non-idle reactor, if $(i,r) \in \mathcal{C}_{id}$ (cf. Eq. (4.23)). This is the case if a candidate CV of a non-idle reactor is not used in any closed loop.

This discussion reveals the important property that the submodels of non-idle reactors and controllers are independent of the submodels of idle reactors and controllers. However, the inverse is not true. The submodels of idle reactors and controllers depend on the submodels of non-idle reactors and controllers, as shown by Eq. (4.30).

4.2 Problem formulation

Having introduced the closed-loop model (4.7), constraints (4.11) for decentralized control structure selection and structural constraints (4.19), (4.21), we present a problem formulation for simultaneous closed-loop reactor network synthesis in this section. We start by formulating an eigenvalue constraint for the closed-loop reactor network synthesis problem and then present the problem formulation.

https://doi.org/10.51202/9783186952035 - Generiert durch IP 216.73.216.36, am 18.01.2026, 21:11:22. © Urheberrechtlich geschützter Inhalt. Ohne gesonderte Erlaubnis ist jede urheberrechtliche Nutzung untersagt, insbesondere die Nutzung des Inhalts im Zusammenhang mit, für oder in KI-Systemen, KI-Modellen oder Generativen Sprachmodellen.

4.2.1 Eigenvalue constraint for simultaneous reactor network and control system synthesis

Let $J_{tot} \in \mathbb{R}^{(n_x+n_e)\times(n_x+n_e)}$ be the Jacobian matrix of the closed-loop system (4.7) with respect to the states x and e. Since u are intermediate variables defined by Eq. (4.7c), J_{tot} can be calculated straightforwardly by using the chain rule.

The right hand side of Eqs. (4.7a), (4.7b) $f_i(\cdot)$ and $\phi_{i',r'} - \bar{y}_{i',r'}$ are used to define

$$F_{id}(\cdot) := (\underbrace{\cdots, f_i(\cdot)^T, \cdots}_{i\in\mathcal{I}_{id}}, \underbrace{\cdots, \phi_{i',r'}(\cdot) - \bar{y}_{i',r'}, \cdots}_{(i',r')\in\mathcal{C}_{id}})^T, \tag{4.32a}$$

$$F_{nid}(\cdot) := (\underbrace{\cdots, f_i(\cdot)^T, \cdots}_{i\in\mathcal{I}_{nid}}, \underbrace{\cdots, \phi_{i',r'}(\cdot) - \bar{y}_{i',r'}, \cdots}_{(i',r')\in\mathcal{C}_{nid}})^T. \tag{4.32b}$$

$F_{id}(\cdot)$ and $F_{nid}(\cdot)$ refer to the state functions of all idle or non-idle reactors and controllers, respectively. Let J_{id} and J'_{id} be the Jacobian matrices of $F_{id}(\cdot)$, J'_{nid} and J_{nid} be the Jacobian matrices of $F_{nid}(\cdot)$ defined as

$$\begin{aligned} J_{id} &:= \frac{\partial F_{id}(\cdot)}{\partial(x_{id}^T, e_{id}^T)^T}, \quad J'_{id} := \frac{\partial F_{id}(\cdot)}{\partial(x_{nid}^T, e_{nid}^T)^T}, \\ J_{nid} &:= \frac{\partial F_{nid}(\cdot)}{\partial(x_{nid}^T, e_{nid}^T)^T}, \quad J'_{nid} := \frac{\partial F_{nid}(\cdot)}{\partial(x_{id}^T, e_{id}^T)^T}. \end{aligned} \tag{4.33}$$

From the discussion in Section 4.1.5, we can conclude that

$$J'_{nid} = 0.$$

Hence, after possible reordering of the sequence of subsystems, we obtain the Jacobian matrix J_{tot} of the closed-loop reactor network

$$J_{tot} = \begin{pmatrix} J_{id} & J'_{id} \\ 0 & J_{nid} \end{pmatrix} \tag{4.34}$$

as an upper-triangular matrix.

Because idle reactors and controllers will not be physically realized, only the eigenvalue spectrum of J_{nid} is of interest for design. Hence, we propose the eigenvalue constraint

$$\alpha_{J_{nid}}(x, e, u, \psi_c) \leq -c \tag{4.35}$$

for the design of closed-loop reactor networks. $c > 0$ is a predefined constant, which refers to the required response speed of the designed system (cf. Eq. (2.17)). $\alpha_{J_{nid}}(x, e, u, \psi_c)$ denotes the spectral abscissa of matrix $J_{nid}(x, e, u, \psi_c)$, the Jacobian of the submodels comprising only the non-idle reactors and controllers. Note that the analytical expression of J_{nid} can be obtained from its definition in Eqs. (4.7), (4.33).

We stress that constraint (4.35) is in general discontinuous, because the size of matrix J_{nid} depends on the number of non-idle reactors and controllers, which may be inferred from the values of $\bar{q}$ and K_v (elements of the decision variables ψ_c). Analyzing the continuity of constraint (4.35) rigorously is a challenging task. For illustration, consider that all controllers in the closed-loop model are idle ($K = 0$), then the closed-loop reactor network design problem reduces to the open-loop case, which is discussed in Section 3.2.2. In this case, the solution will be exactly at a discontinuous point of function $\alpha_{J_{nid}}(\cdot)$ in Eq. (4.35), if idle reactors appear in the final design.

https://doi.org/10.51202/9783186952035 - Generiert durch IP 216.73.216.36, am 18.01.2026, 21:11:22. © Urheberrechtlich geschützter Inhalt. Ohne gesonderte Erlaubnis ist jede urheberrechtliche Nutzung untersagt, insbesondere die Nutzung des Inhalts im Zusammenhang mit, für oder in KI-Systemen, KI-Modellen oder Generativen Sprachmodellen.

Because of the difficulty to treat discontinuous constraints by numerical optimization, one should not directly use Eq. (4.35). Following the idea of Proposition 3.2.2, we propose a constraint equivalent to Eq. (4.35), which is continuous but may still be non-smooth at certain points. Hence, it can be treated more easily by standard numerical optimization methods.

To this end, we define

$$\bar{J} := J_{tot} - MH, \tag{4.36}$$

where $M \in \mathbb{R}$ is a sufficiently large positive constant. $H \in \mathbb{R}^{(n_x+n_e)\times(n_x+n_e)}$ has the same dimension as J_{tot} and is defined by

$$H := I - diag(z_1\, I_1, \cdots, z_n\, I_N, z_{1,1}, \cdots, z_{N,n_c^N}),$$

where $I \in \mathbb{R}^{(n_x+n_e)\times(n_x+n_e)}$, $I_i \in \mathbb{R}^{n_{x_i}\times n_{x_i}}$, $i = 1, \cdots, N$, denote identity matrices. We can now formulate

Proposition 4.2.1. *Assume that $\mathcal{I}_{nid} \neq \emptyset$. If the elements in J_{tot} are bounded and if Eqs. (4.14), (4.16), (4.19), (4.21) hold, then*

$$\alpha_{J_{nid}}(x, e, u, \psi_c) = \alpha_{\bar{J}}(x, e, u, \psi_c, z). \tag{4.37}$$

The proof follows the idea of the proof of Proposition 3.2.2.

With this proposition, we can replace constraint (4.35) equivalently by

$$\alpha_{\bar{J}}(x, e, u, \psi_c, z) \leq -c. \tag{4.38}$$

4.2.2 Problem formulation

We can now formulate an optimization problem for simultaneous reactor and control system synthesis:

$$\min_{x,e,u,\psi_c,z,K_v^+,K_v^-,\hat{K}_v} \quad \phi(x, e, u, \psi_c, z) \tag{4.39a}$$

$$\begin{aligned} s.t.\ 0 = f_i(x_i, \cdots, q_{h(i,k)} g_{h(i,k)}(x_{\bar{h}(i,k)}, u_{\bar{h}(i,k)}, d_{\bar{h}(i,k)}), \cdots, q_{h(i,N)} p_{sys},\\ q_{i,1}, \cdots, q_{i,N}, u_i, d_i), i = 1, \cdots, N, \end{aligned} \tag{4.39b}$$

$$0 = \phi_{i,r}(x_i, d_i) - \bar{y}_{i,r}, i = 1, \cdots, N, r = 1, \cdots, n_c^i, \tag{4.39c}$$

$$0 = e, \tag{4.39d}$$

$$0 = u - \bar{u}, \tag{4.39e}$$

$$-c \geq \alpha_{\bar{J}}(x, e, u, \psi_c, z), \forall \pi_\tau \in [\bar{\pi}_\tau - \Delta\bar{\pi}_\tau, \bar{\pi}_\tau + \Delta\bar{\pi}_\tau], \tag{4.39f}$$

$$\psi_c^U \geq \psi_c \geq \psi_c^L \tag{4.39g}$$

$$[K]_{v,w} = [K^+]_{v,w} - [K^-]_{v,w}, \quad v = 1, \cdots, n_m, w = 1, \cdots, n_c, \tag{4.39h}$$

$$[\hat{K}]_{v,w} = [K^+]_{v,w} + [K^-]_{v,w}, \quad v = 1, \cdots, n_m, w = 1, \cdots, n_c, \tag{4.39i}$$

$$0 \leq [K^+]_{v,w} \perp [K^-]_{v,w} \geq 0, \quad v = 1, \cdots, n_m, w = 1, \cdots, n_c, \tag{4.39j}$$

$$0 \leq [\hat{K}]_{v,w} \perp \sum_{w' \neq w} [\hat{K}]_{v,w'} \geq 0, \quad v = 1, \cdots, n_m, w = 1, \cdots, n_c, \tag{4.39k}$$

$$0 \leq [\hat{K}]_{v,w} \perp \sum_{v' \neq v} [\hat{K}]_{v',w} \geq 0, \quad v = 1, \cdots, n_m, w = 1, \cdots, n_c, \tag{4.39l}$$

https://doi.org/10.51202/9783186952035 - Generiert durch IP 216.73.216.36, am 18.01.2026, 21:11:22. © Urheberrechtlich geschützter Inhalt. Ohne gesonderte
Erlaubnis ist jede urheberrechtliche Nutzung untersagt, insbesondere die Nutzung des Inhalts im Zusammenhang mit, für oder in KI-Systemen, KI-Modellen oder Generativen Sprachmodellen.

$$\begin{bmatrix} z_i \\ \sum\limits_{j=1}^{N} \bar{q}_{i,j} + \sum\limits_{k=1}^{N} \bar{q}_{h(i,k)} > 0 \end{bmatrix} \vee \begin{bmatrix} \bar{z}_i \\ \bar{q}_{i,j} = 0, \forall j = 1, \cdots, N \\ \bar{q}_{h(i,k)} = 0, \forall k = 1, \cdots, N \end{bmatrix}, \forall i = 1, \cdots, N, \tag{4.39m}$$

$$\begin{bmatrix} z_{i,r} \\ \sum\limits_{v=1}^{n_m} [\hat{K}]_{v,\varrho(i,r)} > 0 \end{bmatrix} \vee \begin{bmatrix} \bar{z}_{i,r} \\ \sum\limits_{v=1}^{n_m} [\hat{K}]_{v,\varrho(i,r)} = 0 \end{bmatrix}, \forall i = 1, \cdots, N, r = 1, \cdots, n_c^i, \tag{4.39n}$$

$$z_{i,r} \leq z_i, \quad r = 1, \cdots, n_c^i, i = 1, \cdots, N, \tag{4.39o}$$

$$\begin{bmatrix} z_i \\ \emptyset \end{bmatrix} \vee \begin{bmatrix} \bar{z}_i \\ \sum\limits_{w=1}^{n_c} [\hat{K}]_{v,w} = 0, \forall v \in \Theta_i \end{bmatrix}, i = 1, \cdots, N. \tag{4.39p}$$

$\pi_\tau \in \mathbb{R}^{n_{\pi_\tau}}$ denote uncertain parameters, which are elements of π defined in Eq. (4.3). $\bar{\pi}_\tau$ denote the nominal values of π_τ and $\Delta\bar{\pi}_\tau$ denote the size of the uncertainty region. We have assumed that uncertain parameters π_τ lie always in the region of $[\bar{\pi}_\tau \pm \Delta\bar{\pi}_\tau]$. Upper and lower bounds of the design parameters ψ_c are denoted by ψ_c^U and ψ_c^L, respectively.

Eq. (4.39a) is an economic cost function. Eqs. (4.39b)-(4.39e) define the steady states of the closed-loop model (4.7). Eq. (4.39f) is the so-called robust eigenvalue constraint, which guarantees robust dynamic properties. It requires that the spectral abscissa of matrix $\bar{J}$ is less than $-c$ for all the nearby operating points, derived by varying uncertain parameters π_τ in the uncertain region $[\bar{\pi}_\tau \pm \Delta\bar{\pi}_\tau]$, cf. [119]. The larger c, the faster is the response in the final design. Note that Eq. (4.39f) covers only the dynamic properties of non-idle reactors and controllers. Therefore, the effect of idle reactors and controllers on the dynamic properties of the designed network are ruled out.

Eq. (4.39g) are box constraints on decision parameters ψ_c, which include, for example, bounds on reactor dimension and controller parameters. Eqs. (4.39h)-(4.39l) guarantee a proper decentralized control structure and are reproduced from Eq. (4.11). Eqs. (4.39m), (4.39n) define idle/non-idle reactors and controllers. Eqs. (4.39o), (4.39p) impose structural constraints on individual reactors, controllers and the usage of candidate CV. They are reproduced from Eqs. (4.14), (4.16), (4.19), (4.21) accordingly.

Problem (4.39) formulates a closed-loop reactor network design problem with flexible flowsheet and control structure. Optimal flowsheet and decentralized control structure can be determined simultaneously through solving this single optimization problem. Parametric uncertainties are considered in the proposed formulation such that the final design has robust dynamic properties. Moreover, problem (4.39) determines the optimal number of reactors of each type (PFR or CSTR) as well as the optimal number of PI controllers in the final design.

We note that we are not allowed to choose an arbitrary large value of c, because physical systems always show inherent limitations of the response speed. Choosing a too large c may render the optimization problem (4.39) infeasible.

4.3 Summary

In this section we have formulated optimization problem (4.39) for closed-loop reactor network synthesis, in which both flowsheet and control structure alternatives are design degrees of freedom. Idle reactors and controllers are allowed to appear in the problem

https://doi.org/10.51202/9783186952035 - Generiert durch IP 216.73.216.36, am 18.01.2026, 21:11:22. © Urheberrechtlich geschützter Inhalt. Ohne gesonderte Erlaubnis ist jede urheberrechtliche Nutzung untersagt, insbesondere die Nutzung des Inhalts im Zusammenhang mit, für oder in KI-Systemen, KI-Modellen oder Generativen Sprachmodellen.

formulation. Hence, solving problem (4.39) also determines the optimal number of non-idle reactors and controllers in the final design. The eigenvalue constraint (4.39f) is used to guarantee robust stability and a specified response speed, while complementarity constraints (4.39h)-(4.39l) and disjunctions (4.39m)-(4.39p) ensure a decentralized control system and feasible structural relationships between reactors and controllers.

The presented formulation is directly based on the results of the open-loop reactor network synthesis problem presented in Chapter 3. A decentralized control system is added to the open-loop model (3.20). The idea of formulating eigenvalue constraint (4.38) for the closed-loop reactor network is closely related to Eq. (3.33), and the proposed reformulation strategy (4.36) to treat the discontinuity of eigenvalue constraint (4.35) is derived from Eq. (3.42). Hence, this chapter presenting closed-loop reactor network design can be considered as an extension of the open-loop reactor network design problem shown in Chapter 3.

In contrast to the open-loop reactor network design problem, the complementarity constraints (4.11) and the structural relationships (cf. Section 4.1.4) are novel elements. Eq. (4.11) presents an interesting complementarity-based formulation for decentralized control structure selection. It can be treated by numerical optimization method more efficiently, compared to the integer-based formulations proposed in [124]. This complementarity-based formulation is very promising, because it has the potential to be applied to other control structure selection problems. The structural relationships are proposed to ensure feasible structures, if idle reactors and controllers appear in the final design.

The obtained optimization problem (4.39) is subject to the following limitations. First, the spectral abscissa of the Jacobian matrix (cf. Eq. (4.39f)) is used as a single design criterion for the closed-loop design. It may result in conservative designs, if, for example, fast disturbances appear or the process shows strong nonlinearity. The consideration of other design criteria should be investigated in the future. Second, the specified robustness in Eq. (4.39f) is only defined on the steady states near the nominal operating point. The attractive region U (cf. Theorem 2.3.3) may in certain cases be very small, and if this is the case, any small disturbances will make the system converge to another steady state. Third, the obtained optimization problem (4.39) is very difficult to solve, even for local minima (cf. Section 5.5), because of the combined features of process nonlinearity, non-smoothness of the eigenvalue constraint, complementarity constraints, integer variables, disjunctions and parametric uncertainty. Finally, as in case of the open-loop synthesis problem (cf. Section 3.3), the numerical performance of the big-M reformulation (4.36) and the reformulation strategy regarding Eq. (4.37) should be further investigated.

https://doi.org/10.51202/9783186952035 - Generiert durch IP 216.73.216.36, am 18.01.2026, 21:11:22. © Urheberrechtlich geschützter Inhalt. Ohne gesonderte
Erlaubnis ist jede urheberrechtliche Nutzung untersagt, insbesondere die Nutzung des Inhalts im Zusammenhang mit, für oder in KI-Systemen, KI-Modellen oder Generativen Sprachmodellen.

5 Solution methods

In previous chapters we derived two optimization problems, Eqs. (3.44) and (4.39), which correspond to the open-loop and the closed-loop reactor network design problem, respectively. Both problem formulations combine the features of mixed continuous/integer variables, disjunctions, complementarities, uncertain parameters and eigenvalue constraints. To the author's knowledge, there do not exist numerical solvers in literature or even the open domain that can solve problems with all these features. In this chapter, we will first review each related classes of optimization problems and then propose a two-step hybrid solution strategy for the optimization problems (3.44) and (4.39).

Note that the symbols used in this chapter are not the same as those symbols used in the modeling part of this work in Chapters 3 and 4. To simplify the notation, the meaning of each of the symbols used in this chapter is always explained when they are introduced.

5.1 Discrete-continuous optimization

Mixed-integer nonlinear program (MINLP) and generalized disjunctive program (GDP) are closed related. For this reason, we review these two classes of discrete-continuous optimization problems in one section.

5.1.1 Mixed-integer nonlinear program

MINLP refers to mathematical programs with mixed continuous/discrete variables and nonlinear objective function and/or constraints. A general form of a MINLP is

$$\begin{aligned} \min\ & f(x,y) \\ \text{s.t.}\ & g(x,y) \leq 0, \\ & x \in X, y \in \{0,1\}^{n_y}. \end{aligned} \tag{5.1}$$

The functions $f(x,y)$ and $g(x,y)$ are nonlinear, twice continuously differentiable functions. $X \subset \mathbb{R}^{n_x}$ is a bounded polyhedral set. x is a vector of continuous decision variables, while y is a vector of binary decision variables.

According to the convexity of the objective function and the constraints of a MINLP, we can classify it into either convex or non-convex MINLP. Problem (5.1) is called a convex MINLP, if the functions $f(x,y)$ and $g(x,y)$ are convex, otherwise, it is non-convex [17]. A convex MINLP is easier to solve than a non-convex MINLP. Bonami et al. [22] review the state-of-the-art algorithms for solving convex MINLP and Tawarmalani and Sahinidis [171] review global optimization algorithms for solving non-convex MINLP.

5.1.2 General concepts to solve MINLP

Two concepts that frequently appear in solving convex and non-convex MINLP are relaxation and constraint enforcement [17].

https://doi.org/10.51202/9783186952035 - Generiert durch IP 216.73.216.36, am 18.01.2026, 21:11:22. © Urheberrechtlich geschützter Inhalt. Ohne gesonderte Erlaubnis ist jede urheberrechtliche Nutzung untersagt, insbesondere die Nutzung des Inhalts im Zusammenhang mit, für oder in KI-Systemen, KI-Modellen oder Generativen Sprachmodellen.

Formally, an optimization problem $min\{f_r(x) : x \in S_r\}$ is called a relaxation of another optimization problem $min\{f(x) : x \in S\}$, if $S_r \supseteq S$ and $f_r(x) \leq f(x)$, for all $x \in S$ [17]: A relaxed problem can be obtained by enlarging the feasible set and/or by replacing the original objective function by its lower estimator. From this definition, there may exist many different ways to relax a given MINLP. In practice, however, we are only interested in relaxed problems which can be solved much more easily than the original one. Convex nonlinear programs (NLP) and mixed-integer linear programs (MILP) qualify, because both of them can be solved to global optimality by using local optimization algorithms. For convex MINLP, convex NLP can be generated by relaxing integer variables, and MILP can be generated by applying the outer approximation (OA) method, which linearly relaxes nonlinear convex constraints. The benefits of relaxation are that, the obtained relaxed problem guarantees a lower bound of the original one, and one can use this property in different ways to exclude further investigation of the subproblems such that the optimal solution of the original MINLP can be found in the end.

Constraint enforcement refers to a procedure which excludes solutions that are feasible to the relaxed problem but not feasible to the original MINLP [17]. Constraint enforcement can be accomplished by either relaxation refinement or branching [17]. Because solving a relaxed problem may result in solutions which are not feasible for the original one, the goal of constraint enforcement is to exclude these solutions such that the algorithm can eventually converge to a true solution of the original one. The method of relation refinement usually tightens the relaxation by adding extra inequalities, while the method of branching either branches an integer variable into 0 or 1, or branches the feasible domain of a continuous variable into separate sub-domains. Branching continuous variables requires a division of the feasible domain X of continuous variables into (typically) two sub-domains. Lower bounds for these sub-domains, which do not include the solution of the original problem, will eventually become larger than the upper bound[1] determined by the algorithm. When this happens, these sub-domains can be excluded from further search. Also by successive branching, sub-domains become smaller and smaller, one of them finally indicate the location of the global minimum.

5.1.3 Solution methods for convex MINLP

Methods to solve convex MINLP include branch and bound (B&B) [31, 58], outer approximation (OA) [36, 42], generalized Bender's decomposition (GBD) [52] and extended cutting plane methods [183]. These approaches generally rely on successively solving multiple relaxed convex NLP (e.g., B&B method) or relaxed MILP (e.g., OA method). Comprehensive reviews of convex MINLP methods can be found in [17, 45, 128, 173].

For the sake of brevity, we only review the fundamentals of the B&B method[2]. Original, the B&B method aimed at solving MILP [31]. Later the method was extended to the convex nonlinear case [23, 58, 97]. In B&B, branching is done in the feasible space of integer variables, i.e., integer variables are fixed to either 0 or 1. A sequence of relaxed subproblems, which are convex NLP[3], are solved to global optimality by using a local

---

[1] This upper bound is a feasible point of the original MINLP, but it is not necessarily optimal.

[2] In comparison to the terminology "spatial branch and bound" (sB&B) used in the literature of global optimization, we use B&B to explicitly refer to solution algorithms for convex MINLP, which branch only integer variables into 0 or 1.

[3] The NLP derived from a convex MINLP are convex.

https://doi.org/10.51202/9783186952035 - Generiert durch IP 216.73.216.36, am 18.01.2026, 21:11:22. © Urheberrechtlich geschützter Inhalt. Ohne gesonderte Erlaubnis ist jede urheberrechtliche Nutzung untersagt, insbesondere die Nutzung des Inhalts im Zusammenhang mit, für oder in KI-Systemen, KI-Modellen oder Generativen Sprachmodellen.

NLP solver. The B&B method starts by solving a root subproblem, in which all integer variables y are relaxed by $0 \leq y \leq 1$. Since, in general, discrete values of integer variables can not be a solution of the root subproblem, a tree search of integer variables, namely branching, is performed. Solving relaxed subproblems generates lower bounds, which can be used to fathom subproblems. These subproblems do not have to be branched further. B&B algorithms are suitable for optimization problems, in which there are not too many integer variables and each subproblem can be solved efficiently.

5.1.4 Solution methods for non-convex MINLP

Solving non-convex MINLP is more challenging than solving convex MINLP, because it is not straightforward to obtain valid lower bounds for the purpose of fathoming. Using the relaxation strategy applied for solving convex MINLP, the generated subproblems are generally non-convex. Valid lower bounds can only be produced if the subproblems are solved to global optimality. This is problematic, because solving non-convex subproblems to global optimality is as difficult as solving the original non-convex MINLP. For this reason, new relaxation strategies are proposed in literature, which result in relaxed convex subproblems, e.g., in convex NLP, MILP. These convex subproblems can be solved to global optimality efficiently by using local solvers; therefore, valid lower bounds can be obtained for fathoming. Note that the relaxation strategies used to solve non-convex MINLP are closely related to global optimization methods [46]. A review of global optimization methods can be found in, e.g., [47]. For solving non-convex MINLP by global algorithms, we refer to the textbook [171].

Two procedures are essential for solving non-convex MINLP globally. The first procedure refers to the construction of convex relaxations. This can be done by using global optimization methods for factorable functions. An objective function or a constraint is called factorable, if it can be expressed as the sum of products of unary functions of a finite set $\{sin, cos, exp, log, |\cdot|\}$, whose arguments are variables, constraints, or other functions, which are in turn factorable [17]. Different formulations have been proposed to transform the unary functions and the operators of $\{+, \times, \div, \hat{}\ \}$ into relaxed convex forms, including LP relaxation for monomials of odd degree [100], or convex hull relaxation for the products of two variables [3]. Using these formulations, non-convex MINLP can be ultimately relaxed to convex subproblems, which can be solved to global optimality.

The second procedure partitions the feasible set by branching integers (to be either 0 or 1) and the feasible region of continuous variables. The global optimization community refers to this procedure as spatial branch and bound (sB&B). Branching integers is the same as discussed before in the context of the B&B method for convex MINLP. Partitioning the feasible region of continuous variables means to spatially divide the feasible region of continuous variables into two subsets. The derived subsets are typically disjoint from each other to avoid searching optimal solutions in the same region. Branching and partitioning yield two or more subproblems, which have smaller feasible sets and can be relaxed again by using the same strategy as before. Each derived subproblem is solved to generate tighter lower bounds and to update the upper bound. A subproblem can be fathomed, if its lower bound is larger than the algorithm's estimation of the upper bound for the original MINLP. This procedure is repeated until the gap between the upper and lower bounds lies in a certain tolerance interval.

https://doi.org/10.51202/9783186952035 - Generiert durch IP 216.73.216.36, am 18.01.2026, 21:11:22. © Urheberrechtlich geschützter Inhalt. Ohne gesonderte Erlaubnis ist jede urheberrechtliche Nutzung untersagt, insbesondere die Nutzung des Inhalts im Zusammenhang mit, für oder in KI-Systemen, KI-Modellen oder Generativen Sprachmodellen.

5.1.5 Generalized disjunctive programming

Generalized disjunctive programming (GDP) [94, 95, 140] is closely related to MINLP. GDP is a generalized from of disjunction programing [13] and represents a higher-level representation of MINLP problems [173]. The basic idea of GDP is to use Boolean ($true$, $false$) and continuous variables to formulate constraints. Any GDP problem can be reformulated as a mixed-integer problem, and any MINLP can be posed in the form of a GDP [57]. However, it is more natural to start modeling with a GDP, as it captures more directly both the qualitative (logical) and quantitative (equations) parts of a problem, and then reformulate it into a mixed-integer problem [177].

A GDP is typically transformed to an MINLP by, e.g., the big-M method [125] or the convex hull method [94, 95], and is then solved by algorithms designed for MINLP. The basic ideas of the big-M method and the convex hull method are to replace the disjunctive constraints by a set of algebraic constraints, which comprise binary and continuous variables. The convex hull method has initially been developed for disjunctions with only linear constraints [13]. Afterwards, it has been extended to the nonlinear convex [94] and non-convex case [95]. As the name “convex hull” indicates, the motivation of the convex hull method is to get the tightest convex relaxation of the feasible region, such that tight lower bounds can be obtained. For more details on this method we refer to the above-mentioned original works.

Here, we briefly present the major idea underlining the big-M method. Let us consider the generalized form of a single disjunction

$$\begin{aligned} &\vee_{i\in D} \begin{bmatrix} Y_i \\ \theta_i(x) \leq 0 \end{bmatrix}, \\ &Y_i \in \{true, false\}. \end{aligned} \tag{5.2}$$

D is a finite index set of i, $x \in \mathbb{R}^{n_x}$ is a vector of continuous variables, $Y_i \in \{true, false\}$ is a Boolean variable, and $\theta_i(x)$ is a vector-valued real function. The disjunction in Eq. (5.2) contains several disjunctive terms $[Y_i; \theta_i(x) \leq 0]$. The symbol $\vee$ refers to a logical operator, which has the meaning of logical “or”. Enforcing Eq. (5.2) requires that at least one disjunctive term is true, i.e., that there exists at least one i^* such that $Y_{i^*} = true$ and $\theta_{i^*}(x) \leq 0$.

The big-M method reformulates Eq. (5.2) by using binary variables into the equivalent form

$$\theta_i(x) \leq M_i(1 - y_i), \tag{5.3a}$$

$$\sum_{i\in D} y_i \geq 1, \tag{5.3b}$$

$$y_i \in \{0, 1\}. \tag{5.3c}$$

M_i is a sufficiently large upper bound for constraint $\theta_i(x)$, and y_i is an introduced binary variable. When $y_i = 1$, the inequality constraint (5.3a) becomes $\theta_i(x) \leq 0$. When $y_i = 0$, the inequality constraint (5.3a) becomes redundant, because it is satisfied for sufficiently large M_i. Constraint (5.3b) guarantees that at least one binary variable equals 1. In contrast to Eq. (5.2), Eq. (5.3) contains only algebraic constraints depending on binary and continuous variables. Eq. (5.3) can be treated by MINLP algorithms straightforwardly.

https://doi.org/10.51202/9783186952035 - Generiert durch IP 216.73.216.36, am 18.01.2026, 21:11:22. © Urheberrechtlich geschützter Inhalt. Ohne gesonderte
Erlaubnis ist jede urheberrechtliche Nutzung untersagt, insbesondere die Nutzung des Inhalts im Zusammenhang mit, für oder in KI-Systemen, KI-Modellen oder Generativen Sprachmodellen.

Note that Eq. (5.2) also represents a specialized form, which appears frequently in engineering applications, namely

$$\begin{aligned}&\begin{bmatrix} Y_1 \\ \theta_1(x) \le 0 \end{bmatrix} \vee \begin{bmatrix} \overline{Y}_1 \\ \theta_2(x) \le 0 \end{bmatrix}, \\ &Y_1 \in \{true, false\}.\end{aligned} \tag{5.4}$$

Y_1 denotes a Boolean variable, $\bar{Y}_1$ denotes the negation of Y_1. $\theta_1(x)$ and $\theta_2(x)$ are two vector-valued functions. Because either $Y_1 = true$ or $\bar{Y}_1 = true$, Eq. (5.4) represents one and only one disjunctive term to be true. Using y_1 and $1 - y_1$ to represent Y_1 and $\bar{Y}_1$, respectively, Eq. (5.4) can be reformulated to

$$\begin{aligned}&\theta_1(x) \le M_1(1 - y_1), \\ &\theta_2(x) \le M_2 y_1, \\ &y_i \in \{0, 1\}.\end{aligned} \tag{5.5}$$

5.2 Mathematical programs with complementarity constraints

Mathematical programs with complementarity constraints (MPCC) can be formulated as

$$\begin{aligned}\min\ & f(x, y, z) \\ \text{s.t. } & g(x, y, z) \le 0, \\ & h(x, y, z) = 0, \\ & 0 \le y \perp z \ge 0,\end{aligned} \tag{5.6}$$

where $x \in \mathbb{R}^m$, $y \in \mathbb{R}^p$ and $z \in \mathbb{R}^p$ are decision variables. $g(\cdot)$ and $h(\cdot)$ refer to inequality and equality constraints, respectively. Constraints $0 \le y \perp z \ge 0$ are called complementarity constraints. This notation is a short hand of

$$\begin{aligned}&y \ge 0, \\ &z \ge 0, \\ &y_i = 0 \text{ or } z_i = 0, \forall i = 1, \cdots, p,\end{aligned}$$

where y_i and z_i denote the i-th element of y and z, respectively. In the derived optimization problem (4.39), Eqs. (4.39h)-(4.39l) are complementarity constraints. This section reviews methods to treat optimization problems with complementarity constraints.

The complementarity constraints in MPCC (5.6) can be replaced by, e.g., any of the following equivalent algebraic forms:

$$\begin{aligned}(i)\ & y \ge 0, z \ge 0, y_i z_i \le 0, \forall i = 1, \cdots, p, \\ (ii)\ & y \ge 0, z \ge 0, y^T z = 0, \\ (iii)\ & y \ge 0, z \ge 0, y^T z \le 0.\end{aligned} \tag{5.7}$$

https://doi.org/10.51202/9783186952035 - Generiert durch IP 216.73.216.36, am 18.01.2026, 21:11:22. © Urheberrechtlich geschützter Inhalt. Ohne gesonderte Erlaubnis ist jede urheberrechtliche Nutzung untersagt, insbesondere die Nutzung des Inhalts im Zusammenhang mit, für oder in KI-Systemen, KI-Modellen oder Generativen Sprachmodellen.

MPCC are closely related to mathematical programs with equilibrium constraints (MPEC) [107]. MPCC is actually a special case of MPEC. An MPEC is a constrained optimization problem, in which some or all of its constraints are defined as parametrized variational inequalities (VI) [37]. MPEC take the general form

$$\begin{aligned} \min\ & f'(x', y') \\ \text{s.t.}\ & (x', y') \in Z \subset \mathbb{R}^{m+n}, \\ & y' \in C(x'), \\ & (v' - y')^T F(x', y') \geq 0, \forall v' \in C(x'), \end{aligned} \tag{5.8}$$

where $x' \in \mathbb{R}^m$ and $y' \in \mathbb{R}^n$. Z denotes a feasible subset in $\mathbb{R}^{m+n}$. $C : \mathbb{R}^m \rightarrow \mathbb{R}^m$ is a set-valued function with closed convex values, i.e., $\forall x'$, $C(x')$ denotes a subset in $\mathbb{R}^n$ which is closed and convex. Constraints $(v' - y')^T F(x', y') \geq 0$, $y' \in C(x')$, $\forall v' \in C(x')$, are the so-called variational inequalities (VI), which are parametrized by x'.

Under some assumptions on the VI, i.e., if $C(x') = \mathbb{R}^m_+$, MPEC (5.8) can be reformulated to an equivalent MPCC [107]. From this perspective, it is not surprising that any MPCC can also be reformulated backwards into a MPEC [16]. Because MPEC is more general than MPCC, MPEC are generally more difficult to be solved. For a comprehensive discussions of MPEC, we refer to the textbook [107].

In this work, we will focus on MPCC only. Properties of MPCC and relevant solution methods will be discussed later. General reviews of MPCC are provided in [43, 98, 145]. A good introduction about the fundamentals of MPCC and its relationship to NLP can be found in the introduction section of [178]. The material presented here follows the discussion in [178].

5.2.1 MPCC versus NLP

Although any MPCC can be reformulated into an NLP by applying, e.g., Eq. (5.7), MPCC is different from NLP, because the NLP reformulation resulting from application of Eq. (5.7) does not fulfill regularity conditions, which are typically assumed for solving NLP. In particular, Mangasarian-Fromovitz constraint qualification (MFCQ) (and thus the stronger linear independence constraint qualification (LICQ)) is violated at all feasible points of MPCC. Because constraint qualifications are needed to prove the convergence of standard NLP algorithms, violation of constraint qualification means that the feasible set of MPCC is ill-posed. Therefore, applying NLP algorithms directly to solve MPCC becomes problematic.

Let us try to demonstrate this through a simple example taken from [178], which shows that local minima of the original MPCC do not satisfy the KKT condition of the resulting NLP.

Example 5.1.

$$\begin{aligned} \min_{y_1, z_1 \in \mathbb{R}}\ & z_1 - y_1 \\ s.t.\ & y_1^2 + z_1^2 - 2z_1 \leq 0, \\ & 0 \leq y_1 \perp z_1 \geq 0. \end{aligned} \tag{5.9}$$

It is easy to see that the feasible set of MPCC (5.9) is

$$\mathcal{F} = \{y_1 = 0, z_1 \in [0, 1]\} \subset \mathbb{R}^2,$$

https://doi.org/10.51202/9783186952035 - Generiert durch IP 216.73.216.36, am 18.01.2026, 21:11:22. © Urheberrechtlich geschützter Inhalt. Ohne gesonderte Erlaubnis ist jede urheberrechtliche Nutzung untersagt, insbesondere die Nutzung des Inhalts im Zusammenhang mit, für oder in KI-Systemen, KI-Modellen oder Generativen Sprachmodellen.

and that its optimal solution is $(y_1^*, z_1^*) = (0, 0)$.

If we use (iii) in Eq. (5.7) to reformulate MPCC (5.9), the NLP

$$\begin{aligned} &\min z_1 - y_1 \\ &s.t.\ y_1^2 + z_1^2 - 2z_1 \leq 0, \\ &\qquad - y_1 \leq 0, \\ &\qquad - z_1 \leq 0, \\ &\qquad y_1 z_1 \leq 0, \end{aligned} \tag{5.10}$$

is obtained.

First, it is not difficult to verify that MFCQ for problem (5.10) is violated at $(y_1^*, z_1^*) = (0, 0)$. *Actually, MFCQ for problem (5.10) requires that there exists a vector* $d \in \mathbb{R}^2$, *such that the gradients of all active inequalities, denoted as* ∇g_{act}, *satisfy* $\nabla g_{act} d < 0$. *However, at point* $(y_1^*, z_1^*) = (0, 0)$, *where all four inequalities are active,*

$$\nabla g_{act} = \begin{bmatrix} 0 & -2 \\ -1 & 0 \\ 0 & -1 \\ 0 & 0 \end{bmatrix},$$

and we can not find such a d. Note that, because LICQ implies MFCQ, LICQ is not fulfilled, either.

Second, we show that the optimal solution of MPCC (5.9), namely $(y_1^*, z_1^*) = (0, 0)$, *does not satisfy the KKT condition of NLP (5.10). In this sense, if we use NLP algorithms, such as sequential quadratic programming (SQP), which are designed to converge to the KKT points of NLP (5.10), such algorithms will not converge to the correct solution* $(y_1^*, z_1^*) = (0, 0)$ *of the original MPCC (5.9). The KKT conditions of NLP (5.10) at* $(y_1^*, z_1^*) = (0, 0)$ *take the form*

$$\begin{aligned} &0 = \begin{bmatrix} -1 \\ 1 \end{bmatrix} + \lambda_1 \begin{bmatrix} 0 \\ -2 \end{bmatrix} + \lambda_2 \begin{bmatrix} -1 \\ 0 \end{bmatrix} + \lambda_3 \begin{bmatrix} 0 \\ -1 \end{bmatrix} + \lambda_4 \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \\ &\lambda_1, \cdots, \lambda_4 \geq 0. \end{aligned} \tag{5.11}$$

$\lambda_1, \cdots, \lambda_4$ *denote Lagrangian multipliers for each of the four constraints in NLP (5.10). It is easy to verify that Eq. (5.11) contains no feasible solutions, because it requires* $\lambda_2 = -1$ *and* $\lambda_2 \geq 0$ *simultaneously. Hence, it is problematic to solve MPCC by directly applying KKT conditions to the reformulated NLP.* □

From this simple example, we see that, because of the avoidance of the regularity properties, we can not use standard necessary optimality conditions (e.g. KKT conditions) of NLP to characterize local minima of MPCC. To characterize local solutions of MPCC, different stationarity concepts and constraint qualifications have been proposed in the literature. Scheel and Scholtes [149] proposed four types of stationarity concepts. Among them, the strongly stationarity concept is the strongest condition. There exist also a variety of specialized constraint qualifications for MPCC. For example, MPCC-LICQ is the most frequently used condition, which is also the easiest one to be verified [178]. For more details, we refer to [107, 149, 186, 187].

https://doi.org/10.51202/9783186952035 - Generiert durch IP 216.73.216.36, am 18.01.2026, 21:11:22. © Urheberrechtlich geschützter Inhalt. Ohne gesonderte Erlaubnis ist jede urheberrechtliche Nutzung untersagt, insbesondere die Nutzung des Inhalts im Zusammenhang mit, für oder in KI-Systemen, KI-Modellen oder Generativen Sprachmodellen.

5.2.2 Solution methods for MPCC

Solution methods for MPCC (5.6) strongly relate to NLP solution methods. They can be classified into regularization (smoothing) methods, penalty methods and methods based on direct/adapted application of particular NLP algorithms, such as interior-point (IP), or sequential quadratic programming (SQP) methods [178]. In this section, we first review the penalty method, the SQP method and the interior-point method for MPCC, and then discuss the regularization method.

Penalty method

The penalty method has originally been proposed to solve NLP. The general idea is to replace the constrained optimization problem by a series of unconstrained problems. Under proper conditions, the solutions of the derived unconstrained problem converge to the solution of the original problem.

The idea of the penalty method can also be applied to solve MPCC: complementarity constraints are transformed into a penalty term in the objective function. Instead of solving the original MPCC (5.6), one solves the NLP

$$\begin{aligned} \min\ & f(x,y,z) + Mp(y,z) \\ \text{s.t.}\ & g(x,y,z) \leq 0, \\ & h(x,y,z) = 0. \end{aligned} \tag{5.12}$$

M is a constant, which is sufficiently large. $p(y,z)$ denotes a penalty term. Different researchers proposed different forms of $p(y,z)$. For example, Luo et al. [108] used $p(y,z) = y^T x$ and Scholtes and Stöhr [153] used a suitable extension of the l_1-penalty term. Interesting to note that, instead of providing convergence proof for specialized penalty terms, Hu and Ralph [72] have proposed conditions on formulating functions $p(y,z)$ to ensure convergence.

Interior point method

The interior point method has also originally been developed to solve NLP. The general idea of the interior point method (also called barrier method) is to use a barrier function such that all iterations satisfy the inequalities of the original problem strictly, i.e., that all solution iterates are located inside the interior of the feasible region.

The interior point method for MPCC was first proposed in [104], where the relaxed barrier problem [178]

$$\begin{aligned} \min\ & f(x,y,z) - \mu \sum_{i=1,\cdots,4,\forall j} ln(s_{i,j}) \\ \text{s.t.}\ & h(x,y,z) = 0, \\ & -g(x,y,z) = s_1, \\ & y = s_2, \\ & z = s_3, \\ & \epsilon \mathbf{e} - Yz = s_4, \end{aligned} \tag{5.13}$$

https://doi.org/10.51202/9783186952035 - Generiert durch IP 216.73.216.36, am 18.01.2026, 21:11:22. © Urheberrechtlich geschützter Inhalt. Ohne gesonderte Erlaubnis ist jede urheberrechtliche Nutzung untersagt, insbesondere die Nutzung des Inhalts im Zusammenhang mit, für oder in KI-Systemen, KI-Modellen oder Generativen Sprachmodellen.

is solved. $Y = diag(y_1, \cdots, y_p) \in \mathbb{R}^{p\times p}$, $\mu > 0$ is the barrier parameter, $\mathbf{e} = (1, \cdots, 1)^T \in \mathbb{R}^p$, $\epsilon > 0$ denotes a relaxation parameter, and $s_1, \cdots, s_4$ are the so-called slash variables. $s_{i,j}$ denotes the j-th element of s_i. $ln(s_{i,j})$, $\forall i, j$, refers to a barrier term, which requires that $s_{i,j} > 0$ during all numerical iterations.

In the above formulation, both μ and ϵ need to be stepwise reduced to zero and a series of problems in the form of Eq. (5.13) have to be solved. Liu and Sun [104] proposed a shortcut procedure to simultaneously reduce $\mu \downarrow 0$ and $\epsilon \downarrow 0$. In particular, the barrier parameter μ is selected to be a fraction of ϵ. Global convergence of the proposed algorithm is proven under certain conditions.

In [138], a similar interior point approach is proposed and the analyzed barrier problem is akin to problem (5.13) [178]. The interior point method proposed in [32] is different from previous approaches. The authors use a two-sided relaxation, in which both complementarity and non-negativity constraints are relaxed.

SQP method

Because of the theoretical differences between MPCC and NLP (cf. Section 5.2.1), a direct application of SQP algorithms designed originally for NLP to solve MPCC seems problematic. However, Fletcher et al. [44] were able to show that an SQP method converges quadratically near a strongly stationary point under mild conditions. In their work the NLP reformulation

$$\begin{aligned} \min\ & f(x,y,z) \\ \text{s.t.}\ & g(x,y,z) \le 0, \\ & h(x,y,z) = 0, \\ & y^T z \le 0, \end{aligned} \tag{5.14}$$

of the original MPCC (5.6) was studied. The authors discovered an equivalence relationship between the strongly stationary conditions of MPCC and the KKT conditions of NLP (5.14). It was proven that the sequence generated by applying the SQP method to NLP (5.14) locally converges to the strongly stationary solutions of MPCC (5.6).

The work [44] is further extended by [5, 6]. Anitescu [5] suggested the so-called elastic mode, which transforms a MPCC into a NLP with additional variables such that it has an isolated stationary point and a local minimum at the solution of the original problem. Anitescu et al. [6] studied the global convergence of this SQP method.

Consequently, the direct application of SQP algorithms to solve MPCC are quite promising. However, this approach can only be successful for MPCC with strongly stationary solutions [178]. Next, we will review regularization (smoothing) approaches, which are not subject to this restriction.

Regularization (smoothing) method

The idea of the regularization or smoothing method is to find a way to enlarge the feasible set of MPCC such that regularity conditions of the derived optimization problems (e.g. MFCQ, LICQ) can be achieved. A sequence of regularized NLP are typically generated to approximate the original MPCC. Instead of solving the original MPCC, one solves the generated sequence of regularized NLP. Under certain conditions it can be proven that the limit points of the solutions of the regularized problems are the correct solutions of the

https://doi.org/10.51202/9783186952035 - Generiert durch IP 216.73.216.36, am 18.01.2026, 21:11:22. © Urheberrechtlich geschützter Inhalt. Ohne gesonderte Erlaubnis ist jede urheberrechtliche Nutzung untersagt, insbesondere die Nutzung des Inhalts im Zusammenhang mit, für oder in KI-Systemen, KI-Modellen oder Generativen Sprachmodellen.

original MPCC. In practice, regularized NLP are often generated by continuously reducing an introduced parameter.

A great advantage of the regularization method is that one can use off-the-shelf NLP solvers. One can directly implement a regularization method without too much programming effort to large instances of MPCC. However, a disadvantage of this type of methods is that, in order to find an approximate solution of the original MPEC, multiple NLP subproblems must be solved. Compared with relaxation-free approaches, such as exact penalty methods, or the direct application of the SQP method, solving a sequence of subproblems leads to inferior numerical performance.

Regularization methods for MPCC can be further classified into methods based on nonlinear complementarity problem (NCP) functions and non-NCP-based methods [178].

Definition 5.2.1. *A function* $\varphi : \mathbb{R}^2 \to \mathbb{R}$ *is called a NCP function, if* $\forall a, b \in \mathbb{R}$

$$\varphi(a,b) = 0 \Leftrightarrow a \geq 0, b \geq 0, ab = 0.$$

Exemplary NCP functions are the minimum function,

$$\varphi^{min}(a,b) = min(a,b),$$

and the Fischer-Burmeister (FB) function,

$$\varphi^{fb}(a,b) = a + b - \sqrt{a^2 + b^2}.$$

With some NCP functions $\varphi(\cdot)$, MPCC (5.6) can be reformulated to

$$\begin{aligned} \min\ & f(x,y,z) \\ \text{s.t.}\ & g(x,y,z) \leq 0, \\ & h(x,y,z) = 0, \\ & \varphi(y_i, z_i) = 0, \forall i = 1, \cdots, p. \end{aligned} \tag{5.15}$$

Problem (5.15) is equivalent to MPCC (5.6) in the sense that the feasible regions and the optimal solutions of two problems are exactly the same. However, because NCP functions are not differentiable at $(0,0)$, NLP algorithms can not be directly applied.

The idea of NCP-based methods is to use a positive parameter $\epsilon \in \mathbb{R}$ to smoothen the NCP function. For example, the minimum function φ^{min} and the FB function φ^{fb} can be smoothened by

$$\begin{aligned} \varphi_\epsilon^{min} &= \frac{1}{2}(a + b - \sqrt{(a-b)^2 + 4\epsilon^2}), \\ \varphi_\epsilon^{fb} &= a + b - \sqrt{a^2 + b^2 + 2\epsilon^2}. \end{aligned}$$

For a given sequence $\epsilon_k > 0$, $\epsilon_k \searrow 0$ as $k \to \infty$, smoothened NCP functions φ_{ϵ_k} can be used to construct the sequence of smoothened MPCC

$$\begin{aligned} \min\ & f(x,y,z) \\ \text{s.t.}\ & g(x,y,z) \leq 0, \\ & h(x,y,z) = 0, \\ & \varphi_{\epsilon_k}(y_i, z_i) = 0, \forall i = 1, \cdots, p. \end{aligned} \tag{5.16}$$

https://doi.org/10.51202/9783186952035 - Generiert durch IP 216.73.216.36, am 18.01.2026, 21:11:22. © Urheberrechtlich geschützter Inhalt. Ohne gesonderte Erlaubnis ist jede urheberrechtliche Nutzung untersagt, insbesondere die Nutzung des Inhalts im Zusammenhang mit, für oder in KI-Systemen, KI-Modellen oder Generativen Sprachmodellen.

Problem (5.16) is parametrized by ϵ_k. Because all functions are sufficiently smooth, it is actually a classical NLP.

Therefore, instead of solving MPCC (5.6), one alternatively solves a sequence of NLP (5.16), which is parametrized by ϵ_k. As $\epsilon_k \searrow 0$, the feasible region of problem (5.16) approximates the feasible region of MPCC (5.6). It can be proven that, under certain conditions, the stationary points of problem (5.16) converge to the stationary points of MPCC. For details, we refer to the original works [38, 74, 192].

Methods based on non-NCP functions use other relaxation schemes to reformulate the complementarity constraints in MPCC. The general procedure is, however, the same as the one used in NCP-based methods. That is, a sequence of relaxed NLP are generated and solved, whose stationary points converge to the stationary points of the original MPCC.

Scholtes [152] proposed the following relaxation scheme to reformulate the complementarity constraint in MPCC (5.6):

$$y \geq 0, z \geq 0, y_i z_i \leq \epsilon, \forall i = 1, \cdots, p. \tag{5.17}$$

He proved that, if MPCC-LICQ holds for MPCC, the accumulation points of the stationary points of the relaxed NLP are C-stationary points of the original MPCC. If, in addition, an approaching subsequence satisfies a second order necessary conditions of MPCC, the accumulation points are M-stationary points. And if in addition, an upper level strict complementarity condition hold, they are B-stationary. The work is further extended in [139]. Theoretical results in these works are mainly about the relationship between the reformulated NLP and the original MPCC, boundedness of Lagrange multipliers, local uniqueness of the solutions and smoothness of the solution mapping. These issues are explored under various assumptions on the original MPCC at local stationary points.

Lin and Fukushima [102] proposed the modified relaxation scheme

$$\begin{aligned} y_i z_i &\leq \epsilon^2, \forall i = 1, \cdots, p, \\ (y_i + t)(z_i + t) &\geq \epsilon^2, \forall i = 1, \cdots, p. \end{aligned}$$

The provided convergence proof is closely related to [152]. In this work, it is shown that LICQ holds for the proposed relaxed problem under certain mild conditions. By considering the limiting behavior of the relaxed problem, the authors of [152] proved, that any accumulation point of the stationary points of the relaxed problems is C-stationary to the original MPCC if MPCC-LICQ holds, and that, if the Hessian matrices of the Lagrangian functions of the relaxed problems are uniformly bounded below in the corresponding tangent space, it is M-stationary.

Lin and Fukushima [101] analyzed another relaxation method. Denote $\mathbf{e} = (1, \cdots, 1)^T \in \mathbb{R}^p$ as a vector containing only 1. Denote $e_j \in \mathbb{R}^p$ as a vector, in which the j-th element is 1 and the other elements are 0. $e_0 = \mathbf{0} \in \mathbb{R}^p$ denotes a null vector. Define

$$e_j^k = \frac{1}{k}\mathbf{e} + k e_j, \forall j = 0, \cdots, p.$$

The proposed relaxation of the complementarity constraint is

$$\begin{aligned} y &\geq 0, \\ (e_j^k - y)^T z &\geq 0, \forall j = 0, \cdots, p. \end{aligned}$$

https://doi.org/10.51202/9783186952035 - Generiert durch IP 216.73.216.36, am 18.01.2026, 21:11:22. © Urheberrechtlich geschützter Inhalt. Ohne gesonderte Erlaubnis ist jede urheberrechtliche Nutzung untersagt, insbesondere die Nutzung des Inhalts im Zusammenhang mit, für oder in KI-Systemen, KI-Modellen oder Generativen Sprachmodellen.

Recently, Steffensen and Ulbrich [160] introduced a new relaxation method for MPCC. Let $\theta(\cdot) : \mathbb{D} \rightarrow \mathbb{R}$ be a function on $\mathbb{D} = [-1, 1]$, which satisfies: (1) $\theta(\cdot)$ is twice continuously differentiable on $[-1, 1]$; (2) $\theta(1) = \theta(-1) = 1$; (3) $\theta'(1) = 1$, $\theta'(-1) = -1$; (4) $\theta''(1) = \theta''(-1) = 0$; (5) θ is strictly convex in $(-1, 1)$. An exemplary $\theta(\cdot)$ is

$$\theta_{ex}(a) = \frac{1}{8}(a^4 + 6a^2 + 3),$$

where $a \in \mathbb{R}$.

By introducing function $\phi : \mathbb{R} \times \mathbb{R} \times \mathbb{R}_+ \rightarrow \mathbb{R}$ defined by

$$\phi(y_i, z_i, \epsilon) = \begin{cases} |y_i - z_i|, \text{if } |y_i - z_i| \geq \epsilon \\ \epsilon\theta(\frac{y_i - z_i}{\epsilon}), \text{otherwise,} \end{cases}$$

Steffensen and Ulbrich [160] proposed a relaxation of the complementarity constraints

$$y \geq 0, z \geq 0, y_i + z_i \leq \phi(y_i, z_i, \epsilon), \forall i = 1, \cdots, p.$$

The authors proved that, limit points of the stationary points of the relaxed NLP are C-stationary, if they satisfy the so-called MPEC-constant rank constraint qualification. Furthermore, they show if a limit point satisfies MPEC-LICQ and the stationary points satisfy a second-order sufficient condition, this limit point is M-stationary.

5.2.3 Relationship to MINLP

Solution methods designed for MPCC can also be applied to solve MINLP. The basic idea is to transform MINLP (5.1) into MPCC by replacing integer constraints $y \in \{0, 1\}^{n_y}$ by complementarity constraints

$$0 \leq y_i \perp (1 - y_i) \geq 0, \forall i = 1, \cdots, n_y. \tag{5.18}$$

This way, the MPCC methods can be directly applied to the resulting problem

$$\begin{aligned} \min\ & f(x, y) \\ \text{s.t. } & g(x, y) \leq 0, \\ & x \in X, \\ & 0 \leq y_i \perp (1 - y_i) \geq 0, \forall i = 1, \cdots, n_y. \end{aligned} \tag{5.19}$$

Because all integer constraints in MINLP (5.1) have been replaced by complementarity constraints, problem (5.19) is called complementarity-based reformulation of MINLP.

Problem (5.19) is a special type of MPCC. There are some works, which discuss solution methods for this type of problems. Herty and Steffensen [63] studied some theoretical issues of the reformulated problem (5.19), i.e., stationary conditions, feasibility, existence and optimality of the limit points of a sequence of stationary points. Baumrucker et al. [16] compared numerical performance of different smoothing methods for problem (5.19) and investigated the wide application of MPCC in chemical engineering. Stein et al. [165] considered the specialized continuous reformulation of integer constraints, which takes the form of

$$\begin{aligned} &(y_i - 0.5)^2 + ((1 - y_i) - 0.5)^2 \leq 0.5, \\ &(y_i - 0.5)^2 + ((1 - y_i) - 0.5)^2 \geq (\frac{1}{\sqrt{2}} - \epsilon)^2, \end{aligned}$$

https://doi.org/10.51202/9783186952035 - Generiert durch IP 216.73.216.36, am 18.01.2026, 21:11:22. © Urheberrechtlich geschützter Inhalt. Ohne gesonderte Erlaubnis ist jede urheberrechtliche Nutzung untersagt, insbesondere die Nutzung des Inhalts im Zusammenhang mit, für oder in KI-Systemen, KI-Modellen oder Generativen Sprachmodellen.

$\forall i = 1, \cdots, n_y$. $\epsilon > 0$ is a parameter for relaxation. The authors also extended the method to treat GDP problems.

A great advantage of solving MPCC (5.19) instead of solving the original MINLP (5.1), is that local minima[4] of the original problem (5.1), which approximate its global minimum, can be calculated in an efficient way. The complementarity-based reformulation (5.19) has the property that one can use local solvers for MPCC to determine the optimal combination of integer variables directly. In particular, one does not need to fix integer values a priori. Local solvers can be initialized randomly for both continuous and integer variables, and the optimal solution will be decided by the applied solver automatically. Furthermore, multiple-start strategies can be applied to obtain a good local minimum. The reformulation of Eq. (5.19) is especially useful, if the original MINLP is of large size and can not be solved by global algorithms in reasonable time.

5.3 Semi-infinite programming

Semi-infinite programming (SIP) considers the optimization problem

$$\min_{x} f(x) \tag{5.20a}$$

$$\text{s.t. } g(x,y) \geq 0, \ \forall y \in T. \tag{5.20b}$$

where $x \in X \subset \mathbb{R}^m$ and $y \in Y \subset \mathbb{R}^n$, $f : \mathbb{R}^m \to \mathbb{R}$, and $g : \mathbb{R}^m \to \mathbb{R}$ have second-order continuous derivatives. $T \subset \mathbb{R}^n$ is a known compact set, which contains an infinite number of elements. For simplicity, additional equality constraints are omitted. Problems of this type arise in a variety of engineering applications. General reviews of SIP can be found in [66, 105, 161, 180].

Problem (5.20) is called an “infinite” program, because the set T contains an infinite number of elements. As a result, there are infinitely many constraints in Eq. (5.20b). Hence, SIP contain a finite number of decision variables, i.e., x and y, but an infinite number of constraints.

Constraint (5.20b) can be reformulated equivalently by

$$0 \leq \min_{y \in T} g(x,y). \tag{5.21}$$

So, we can define the feasible set of SIP (5.20) by

$$\mathcal{F} = \{x \in X \mid 0 \leq \min_{y \in T} g(x,y)\}. \tag{5.22}$$

We also see that, if we denote

$$Q(x) : \min_{y \in T} g(x,y) \tag{5.23}$$

as a parametrized optimization problem, $Q(x)$ is actually an inner (lower-level) optimization problem of the original SIP. Note that, however, to guarantee feasibility of SIP (5.20), $Q(x)$ must be solved to global optimality. Local minima of $Q(x)$ are not sufficient to guarantee the feasibility of the original SIP. Therefore, solving SIP is more challenging than solving NLP.

---

[4] Local minima of a non-convex MINLP refer to the local minima of the derived NLP, in which all integer variables of the original MINLP are fixed to binary values (either 0 or 1).

https://doi.org/10.51202/9783186952035 - Generiert durch IP 216.73.216.36, am 18.01.2026, 21:11:22. © Urheberrechtlich geschützter Inhalt. Ohne gesonderte Erlaubnis ist jede urheberrechtliche Nutzung untersagt, insbesondere die Nutzung des Inhalts im Zusammenhang mit, für oder in KI-Systemen, KI-Modellen oder Generativen Sprachmodellen.

Problem (5.20) is a specialization of a so-called bi-level (BL) optimization problem. A BL optimization problem takes the general form [163]

$$\begin{aligned} &\min_{x,y} f_b(x,y) \\ &\text{s.t.} 0 \leq g_b(x,y), \\ &\quad y \in \arg\min_{y \in T_b} h_b(x,y), \end{aligned} \tag{5.24}$$

where f_b, g_b, h_b have second-order continuous derivatives, T_b is a given compact set. *arg* $\min_{y \in T_b}$ $h_b(x,y)$ denotes a set of points, for which the function $h_b(x,y)$ attains its global minima with respect to $y \in T_b$.

Any SIP (5.20) can be equivalently transformed into a BL optimization problem (5.24) [163], if we set

$$\begin{aligned} f_b(x,y) &= f(x), \\ g_b(x,y) &= g(x,y), \\ h_b(x,y) &= g(x,y), \\ T_b &= T. \end{aligned} \tag{5.25}$$

A more comprehensive review of the relation between SIP and BL optimization problems can be found in [163].

Another more generalized form of SIP is the so-called generalized semi-infinite program (GSIP), in which the set T is not fixed but depends on x. Then, the constraints of a GSIP take the form $g(x,y) \leq 0, \forall y \in T(x)$, where $T(\cdot)$ is a set-valued function, $T(x) \subset \mathbb{R}^m$. The first impression is that GSIP is merely a slight generalization of standard SIP. However, Jongen et al. [80] first indicated that the feasible set of GSIP may not be compact. This makes GSIP significantly differ from SIP. A comparison of GSIP and SIP can be found in [166], while general results about GSIP can be found in [162–164] and the references therein. In this work, we only review the fundamentals of SIP.

A SIP (5.20) is also called an uncertain or a robust optimization problem in applications, because variables x and y can be referred to certain and uncertain design parameters for engineering purposes, respectively. The set T represents then the size of the uncertainty region. Constraint (5.20b) represents a specific design requirement, e.g., feasibility or stability. Satisfying constraint (5.20b) robustly guarantees that the design requirement is fulfilled for any realization of uncertain parameter y in the uncertain region T.

5.3.1 Local solution methods for SIP

Methods for solving SIP (to local minima) can be classified into local constraint reduction methods, discretization methods and exchange methods [66], depending on how the subproblems are generated. An extensive survey can be found in [143]. Several more recent methods are summarized in [161], which are based on specialized techniques of complementarity, lower-level Wolfe duality and semi-smooth approaches. In this work, we review the classical approaches to find local minima of SIP.

Local constraint reduction methods

Like the solution algorithms for NLP, local constraint reduction algorithms for SIP can be characterized by the property whether the algorithm is locally or globally convergent.

https://doi.org/10.51202/9783186952035 - Generiert durch IP 216.73.216.36, am 18.01.2026, 21:11:22. © Urheberrechtlich geschützter Inhalt. Ohne gesonderte Erlaubnis ist jede urheberrechtliche Nutzung untersagt, insbesondere die Nutzung des Inhalts im Zusammenhang mit, für oder in KI-Systemen, KI-Modellen oder Generativen Sprachmodellen.

Local convergence refers to the property that a local minimum can be found, if the initial starting point lies sufficiently close to the local minimum. Global convergence refers to the property that an algorithm can converge to local minima from remote starting points[5]. Local/global convergence for the case of NLP can be found in [60, 99]. Because in practice it is hard to know where the starting points should be selected, global convergence is preferred.

Local constraint reduction methods with local convergence

The main idea of local constraint reduction methods is to locally solve a SIP by solving an alternative finite optimization problem, i.e., an NLP. This goal is achieved by representing the feasible set of the original SIP, which is defined by an infinite number of constraints by a finite number of constraints. The finite number of constraints are derived from the optimality conditions of the inner problem $Q(x)$ in Eq. (5.23). The name "reduction" comes from the fact that the number of constraints is "reduced" from an infinite to a finite number. Local constraint reduction methods with local convergence can be dated back to [65]. We briefly review the development presented there.

Denote point $\bar{x} \in \mathcal{F}$ as a feasible point of SIP (5.20), define

$$T_a(\bar{x}) := \{y \in Y \mid g(\bar{x}, y) = 0\}.$$

$T_a(\bar{x})$ is called the active set of SIP at $\bar{x}$. Note that, $T_a(\bar{x})$ may not necessarily be a finite set. We represent $T_a(\bar{x})$ by

$$T_a(\bar{x}) = \{\bar{y}_k \mid k \in K\} \tag{5.26}$$

using an index set K, which is not necessarily finite. K actually depends on the reference point $\bar{x}$, but we still use $K(\bar{x}) = K$, if not misleading.

In order to control the irregular behavior of $T_a(\bar{x})$, two regularizing conditions are proposed in [65]. One describes the structure of T and the other requires non-degeneracy of the active set $T_a(\bar{x})$.

Condition 5.3.1. *(i) Set T can be described by a finite number of inequalities, i.e.*

$$T = \{y \in \mathbb{R}^n \mid h_l(y) \le 0, \forall l \in L\},$$

where L is a finite index set. $h_l : \mathbb{R}^m \to \mathbb{R}$, $l \in L$, are $\mathcal{C}^2$-functions.

(ii) $\forall y \in T$, define

$$L_a(y) = \{l \in L \mid h_l(y) = 0\}.$$

Assume that $\forall y \in T$, the gradients $\nabla_y h_l(y)$, $l \in L_a(y)$, are linearly independent.

Under Condition 5.3.1, the inner problem $Q(x)$, evaluated at $\bar{x}$, can be denoted as

$$Q(\bar{x}) : \min_y g(\bar{x}, y) \ s.t. \ h_l(y) \le 0, \forall l \in L.$$

Note that, from the definition of $T_a(\bar{x})$, $\bar{y}_k$ are global minima of $Q(\bar{x})$.

The second proposed condition is as follows.

---

[5] One should not be confused with algorithms which attempt to find global minima.

https://doi.org/10.51202/9783186952035 - Generiert durch IP 216.73.216.36, am 18.01.2026, 21:11:22. © Urheberrechtlich geschützter Inhalt. Ohne gesonderte
Erlaubnis ist jede urheberrechtliche Nutzung untersagt, insbesondere die Nutzung des Inhalts im Zusammenhang mit, für oder in KI-Systemen, KI-Modellen oder Generativen Sprachmodellen.

Condition 5.3.2. *For any* $\bar{y}_k \in T_a(\bar{x})$*, a second-order sufficient condition with strict complementarity slackness for* $Q(\bar{x})$ *holds at* $y = \bar{y}_k$*. That is, there exists real numbers (multipliers)* $\bar{\beta}_{k,l} > 0$ *(strict complementarity slackness),* $l \in L_a(\bar{y}_k)$*, such that*

$$\nabla_y g(\bar{x}, \bar{y}_k) + \sum_{l \in L_a(\bar{y}_k)} \bar{\beta}_{k,l} \nabla_y h_l(\bar{y}_k) = 0, \tag{5.27}$$

and $\forall \delta \in H_k / \ \{0\}$*, where*

$$H_k := \{\delta \in \mathbb{R}^n | \delta^T \nabla_y h_l(\bar{y}_k) = 0, l \in L_a(\bar{y}_k)\}, \tag{5.28}$$

we have

$$\delta^T [\nabla_{yy} g(\bar{x}, \bar{y}_k) + \sum_{l \in L_a(\bar{y}_k)} \bar{\beta}_{k,l} \nabla_{yy} h_l(\bar{y}_k)] \delta > 0. \tag{5.29}$$

Condition 5.3.1 and 5.3.2 lead to two important consequences: (i) Set K is finite, i.e., $T_a(\bar{x})$ is a finite set. (ii) It is possible to describe the effect of variations of $\bar{x}$ on the minima $\bar{y}_k$ in $Q(\bar{x})$, as discussed in the following.

The KKT equation system of NLP $Q(\bar{x})$ is,

$$\begin{aligned} 0 =& \nabla_y g(\bar{x}, y_k) + \sum_{l \in L_a(\bar{y}_k)} \beta_{k,l} \nabla_y h_l(y_k), \\ 0 =& h_l(y_k), l \in L_a(\bar{y}_k) \end{aligned} \tag{5.30}$$

with solutions $y_k = \bar{y}_k$ and $\beta_{k,l} = \bar{\beta}_{k,l} > 0$. The Jacobian matrix of Eq. (5.30) leads to the following property

Lemma 5.3.1 (refer to Eq. (3.6) in [65])**.** *Under Condition 5.3.1 and 5.3.2, the Jacobian matrix of Eq. (5.30) with respect to* y_k *and* $\beta_{k,l}$ *at* $y_k = \bar{y}_k$ *and* $\beta_{k,l} = \bar{\beta}_{k,l}$ *is non-singular.*

This Lemma and the Implicit Function Theorem 2.5.1 guarantee that there exists a neighborhood $U_{\bar{x}}$ of $\bar{x}$, such that for every $k \in K$, y_k and $\beta_{k,l}$ are uniquely determined by $x \in U_{\bar{x}}$. In other words, there exist $\mathcal{C}^1$-functions $y_k : U_{\bar{x}} \to \mathbb{R}^n$, $\beta_{k,l} : U_{\bar{x}} \to \mathbb{R}$, $\forall l \in L_a(\bar{y}_k)$ such that

$$\begin{aligned} 0 \equiv& \nabla_y g(x, y_k(x)) + \sum_{l \in L_a(\bar{y}_k)} \beta_{k,l}(x) \nabla_y h_l(y_k(x)), \\ 0 \equiv& h_l(y_k(x)), l \in L_a(\bar{y}_k). \end{aligned}$$

Since T is compact, there is a neighborhood $V_{\bar{x}} \subset U_{\bar{x}}$ such that for every $x \in V_{\bar{x}}$ the global minima of $Q(x)$ are contained in a finite set

$$\{y_k(x) \mid k \in K(\bar{x})\}.$$

This makes it possible to exactly describe the feasible set $\mathcal{F}$ by a finite number of constraints in the neighborhood V_x.

Theorem 5.3.2 (Local constraint reduction, refer to Constraint-Reduction-Lemma in [65])**.** *Let* $\bar{x} \in \mathcal{F}$*, if Condition 5.3.1 and 5.3.2 hold, then there is a neighborhood* $V_{\bar{x}}$ *of* $\bar{x}$*, and uniquely defined* $\mathcal{C}^1$*-functions* $y_k : V_{\bar{x}} \to \mathbb{R}^n$*,* $k \in K$ *(a finite set), so that*

$$\mathcal{F} \cap V_{\bar{x}} = \{x \in V_{\bar{x}} \mid g(x, y_k(x)) \geq 0, k \in K(\bar{x})\}.$$

https://doi.org/10.51202/9783186952035 - Generiert durch IP 216.73.216.36, am 18.01.2026, 21:11:22. © Urheberrechtlich geschützter Inhalt. Ohne gesonderte Erlaubnis ist jede urheberrechtliche Nutzung untersagt, insbesondere die Nutzung des Inhalts im Zusammenhang mit, für oder in KI-Systemen, KI-Modellen oder Generativen Sprachmodellen.

The above theorem shows that, the feasible set $\mathcal{F}$, which is originally described by an infinite number of constraints in Eq. (5.22), can be described now by a finite number of constraints. Hence, instead of treating an infinite number of constraints to identify local minima $\bar{x}$ of the original SIP, one can alternatively solve the finite optimization problem

$$\begin{aligned} &\min_{x \in V_{\bar{x}}} f(x) \\ &s.t.\ g(x, y_k(x)) \geq 0, k \in K(\bar{x}). \end{aligned} \tag{5.31}$$

Furthermore, because $y_k(x)$ and $\beta_{k,l}(x)$, $l \in L_a(\bar{y}_k)$, are implicitly defined by Eq. (5.30), to numerically compute an optimal solution $\bar{x}$ of SIP (5.20), under several additional conditions [65], one can solve the following KKT system of Problem (5.31) by using e.g. the Newton method:

$$\begin{aligned} 0 =& \nabla_x f(x) - \sum_{k \in K} \lambda_k \nabla_x g(x, y_k), \\ 0 =& g(x, y_k), k \in K \\ 0 =& \nabla_y g(x, y_k) + \sum_{l \in L_a(\bar{y}_k)} \beta_{k,l} \nabla_y h_l(y_k), k \in K \\ 0 =& h_l(y_k), l \in L_a(\bar{y}_k), k \in K. \end{aligned} \tag{5.32}$$

Eq. (5.32) is obtained by explicitly formulating the KKT conditions of problems (5.31) and Eq. (5.30) together. $\lambda_k \geq 0$, $k \in K$, denote multipliers for the inequalities in problem (5.31). Eq. (5.32) consists of an equal number of unknown variables (i.e. x, λ_k, y_k, $\beta_{k,l}$) and nonlinear equations. If initial values of unknown variables are chosen sufficiently close to the local minimum $\bar{x}$ of the original problem, i.e., the inner optimal solution $\bar{y}_k$ and the corresponding multipliers $\bar{\lambda}_k$, $\bar{\beta}_{k,l}$, the Newton method applied to the equation system (5.32) can be used to converge to the local minima $\bar{x}$ of the original SIP. The convergence rate should be at least superlinear.

In summary, the main idea of the local constraint reduction method is that, under regular conditions, local minima of function $g(x, y)$ with respect to $y \in T$ can be expressed by a finite number of $\mathcal{C}^1$-functions $y_k(x)$, $k \in K$(a finite set), in a neighborhood V_x of x. Thus, one can reformulate constraint (5.20b) equivalently by a finite number of constraints $g(x, y_k(x)) \geq 0$, $k \in K$ and obtain a locally equivalent NLP in the form of (5.31). Note that, however, the reformulation (5.31) is only valid in the neighborhood V_x of a reference point x. Beyond this neighborhood, an adaptation of the index set $K(x)$ may be needed.

Local reduction methods with global convergence

The procedure presented above has a significant drawback, since one needs a good initial guess and also a correct guess of the index set K when constructing and solving Eq. (5.32). Algorithms for SIP with global convergence try to avoid this drawback.

Globally convergent algorithms for SIP, also called globalized algorithms for short, are adapted from globalized algorithms for NLP. To the author's knowledge, [60] was one of the first proposed globalized algorithms for NLP. A recent review on this topic can be found in [99]. The basic idea to guarantee global convergence for NLP is to use penalty functions so that a sequence of iteration points, which successively reduce the value of the penalty function, can approach a local minimum of the original NLP. The globalized algorithm presented in [60] is based on a constructed L_1 exact penalty function. The author

https://doi.org/10.51202/9783186952035 - Generiert durch IP 216.73.216.36, am 18.01.2026, 21:11:22. © Urheberrechtlich geschützter Inhalt. Ohne gesonderte Erlaubnis ist jede urheberrechtliche Nutzung untersagt, insbesondere die Nutzung des Inhalts im Zusammenhang mit, für oder in KI-Systemen, KI-Modellen oder Generativen Sprachmodellen.

introduced a step-wise procedure, in which the step sizes are determined to maintain a monotone decrease of the constructed penalty function. It is proven that this procedure converges to a KKT point of the original NLP, starting from a remote initial point.

Since an SIP can be locally represented by NLP (5.31), globalized algorithms for SIP are similar to the ones designed for NLP. However, because the index set $K(x)$ is unknown for SIP, a procedure to update the index set $K(x)$ must be integrated in addition.

In the following, we outline a globalized algorithm for SIP [30] and demonstrate that it is similar to the globalized algorithm proposed for NLP [60]. Note that, there are also other globalized algorithms for SIP [131, 169, 170, 181]. These works differ from the above one [30] in different ways, e.g., the construction of penalty functions, the update of the Hessian matrix, or the calculation of decent directions/step-sizes. A comprehensive comparison of globalized algorithms for SIP can be found in [143].

Coope and Watson [30] constructed a L_1 exact penalty function for SIP. Since SIP (5.20) is locally equivalent to NLP (5.31), the constructed L_1 exact penalty function takes the same form as the one used in [60]:

$$\tilde{\theta}_r(x) = f(x) + r \sum_{k \in K(x)} [g(x, y_k(x))]_+, \tag{5.33}$$

where $r > 0$ denotes a penalty constant. $K(x)$, as it is defined before, denotes a finite set containing the global minima of $Q(x)$. $[g(x, y_k(x))]_+$ is equal to $g(x, y_k(x))$, if $g(x, y_k(x)) \geq 0$. Otherwise, $[g(x, y_k(x))]_+ = 0$. Global minima of $Q(x)$ have been formulated by $\mathcal{C}^1$-functions $y_k(x)$ such that the penalty function $\tilde{\theta}_r(x)$ only depends on x.

We use j, $j = 1, 2, \cdots$, to denote the major iterations of the algorithm proposed in [30]. Denote x_j as the j-th iterate. Similarly, denote p_j as the search direction and s_j as the step length for the j-th iteration. The proposed algorithm is similar to the procedure in [60], except that the set $K(x_j)$ must be updated at each iterations, because the global minima of $Q(x_j)$ depend on the current iteration point x_j. In their approach, p_j is selected by solving a quadratic program, and it is proven that the p_j obtained is always a decent direction of the penalty function $\tilde{\theta}_r(x)$. The step length s_k is chosen to satisfy

$$\frac{\tilde{\theta}_r(x_j + s_j p_j) - \tilde{\theta}_r(x_j)}{s_j \partial_{p_j} \tilde{\theta}_r(x_j)} \geq \epsilon, \forall j, \tag{5.34}$$

where $\partial_{p_j} \tilde{\theta}_r(x_j)$ denotes the directional derivative of the penalty function $\tilde{\theta}_r(x_j)$ along direction p_j. $\epsilon > 0$ is a given small constant. It is shown [30] that, under several reasonable conditions, a limit point of the sequence x_j is a stationary point of the original SIP.

Discretization method

The basic idea of discretization methods is to minimize the objective function of SIP (5.20), which is subject to a carefully selected finite subset of the infinite set of original constraints. By increasing the cardinality of the finite subset, i.e., by intensifying the discretization, a sequence of finite problems (NLP) can be solved, whose solutions approach the solution of the original SIP. Hence, discretization methods solve a sequence of discretized subproblems

$$\begin{aligned} &\min_{x,y} f(x) \\ &\text{s.t.} g(x, y) \geq 0, \forall y \in T_i, \end{aligned} \tag{5.35}$$

https://doi.org/10.51202/9783186952035 - Generiert durch IP 216.73.216.36, am 18.01.2026, 21:11:22. © Urheberrechtlich geschützter Inhalt. Ohne gesonderte Erlaubnis ist jede urheberrechtliche Nutzung untersagt, insbesondere die Nutzung des Inhalts im Zusammenhang mit, für oder in KI-Systemen, KI-Modellen oder Generativen Sprachmodellen.

where $T_i \subset T$ is a finite subset of T. $i = 1, \cdots, +\infty$ is an index for the generated sequence of subproblems. Since problem (5.35) contains only a finite number of constraints, it can be solved by NLP optimization algorithms.

A key task of discretization methods is to find a way to construct the sequence T_1, T_2, $\cdots$, so that solutions of subproblems (5.35) converge to the real solutions of the original SIP (5.20) efficiently. There are generally two ways to do this: a pre-determined way and an adapted way [96]. The pre-determined way discretizes the set T in a pre-defined manner, which does not change during the solution of individual subproblems. Because the set T_i may include too many unnecessary discretized points, this approach often leads to big subproblems which are costly to solve. The adapted way, however, adapts the discretization grids during the solution of the subproblems smartly, such that only a lower number of discretization points are included in T_i. As a result, the subproblems can be solved more efficiently. We refer to [64, 130, 136] for further discussions on adapted discretization methods. Convergence proofs of discretization-based methods for SIP can be found [135, 142, 167].

A great advantage of discretization methods is that they converge robustly under mild conditions (compared to the local constraint reduction method) and one can directly use off-the-shell NLP solvers. However, because convergence is typically guaranteed only for dense discretization grids and in a limiting manner, the method experiences rapid growth in the cardinality of the discretization set T_i, such that the resulting NLP are costly to solve [18].

Two-phase hybrid method

The local constraint reduction method and the discretization method mentioned before both have disadvantages. The local constraint reduction method globally convergences under several assumptions, which have to be satisfied during iterations. In practice, these assumptions may fail and as a result the entire algorithm may fail. The discretization method can converge globally under weaker conditions and therefore behaves more robustly in practice. However, a fine discretization leads to a large and ill-conditioned NLP and the convergence rate of the discretization method near a local minimum is linear [182]. For this reason, a two-phase hybrid method [59] has been proposed in the literature. This type of methods tries to combine the benefits of the local constraint reduction method and the discretization method.

The basic idea of two-phase hybrid methods is to use a discretization method to globally approach a local minimum of SIP from a remote starting point (the first phase), and then to use a local constraint reduction method, which is initialized using the result of the first phase, to exactly identify the local minimum of the original SIP (the second phase). The method is based on the understanding that the solution from the first phase are typically close to a local minimum of the original SIP, and therefore quantifies suitable initial approximations. A good guess of index set K can also often be obtained solving the first phase problem. These approximations will be fed into the second phase. In case that a bad approximation is obtained from the first phase, a finer discretization grid can be applied. Because of these properties, a more robust convergence behavior of the second phase can be expected in practice. Numerical experiments of two-phase hybrid methods can be found in [137].

https://doi.org/10.51202/9783186952035 - Generiert durch IP 216.73.216.36, am 18.01.2026, 21:11:22. © Urheberrechtlich geschützter Inhalt. Ohne gesonderte Erlaubnis ist jede urheberrechtliche Nutzung untersagt, insbesondere die Nutzung des Inhalts im Zusammenhang mit, für oder in KI-Systemen, KI-Modellen oder Generativen Sprachmodellen.

An intrinsic difficulty of two-phase hybrid methods is the determination of a proper discretization resolution in the first phase [143]. If a sequence of refined discretization resolutions are used in the first phase, one also has to determine a proper switch from the first to the second phase. Coarse grids in the first phase may lead to bad approximations of the local minimum, while fine discretizations will generate computationally demanding subproblems. From numerical experiments, it is observed that for lower dimensional problems the solution of a discretized problem on a coarse grid provides a good starting for the second phase [143]. But for higher dimensional problems, when the inner problem has multiple minimizers, it may be quite difficult to guess the correct number of constraints, namely the set K, which should be included in the reduced problem (5.31) of the second phase [143].

5.3.2 Global solution methods for SIP

In contrast to the local methods for SIP discussed before, global solution algorithms for SIP aim to find a global minimum of SIP (5.20). To the author's knowledge, there are only few works on this topic, e.g., algorithms based on interval analysis [18] and algorithms based on relaxing the right hand side [115]. For GSIP, global algorithms are also proposed in literature [116]. In this work, we briefly review the algorithm based on interval analysis [18].

An algorithm based on interval analysis

Bhattacharjee et al. [18] used a spatial branch-and-bound (sB&B) framework to generate a convergent sequence of upper and lower bounds of the global minimum of SIP (5.20). The algorithm converges in a finite number of iterations to ϵ-optimality[6].

An upper-bounding problem is generated by replacing the infinite number of constraints (5.20b) with a finite number of tightened constraints. Assume that the set T is a Cartesian product of intervals, i.e., $T = T_1 \times \cdots T_n$, where each T_i, $i = 1, \cdots, n$, is an interval in $\mathbb{R}$. Denote $T_\tau \in \mathbb{R}^n$ as any interval subset of T. Note that it is allowed to choose $T_\tau = T$. Define function

$$\bar{g}(x, T_\tau) := \{g(x, y) \mid y \in T_\tau\} = [g^l(x, T_\tau), g^u(x, T_\tau)],$$

where $g^l(x, T_\tau)$ denotes the lower-bound function and $g^u(x, T_\tau)$ denotes the upper-bound function of $g(x, y)$ with respect to $y \in T_\tau$. The domain of function $\bar{g}(x, T_\tau)$, as it is defined, is a Cartesian production of values in $\mathbb{R}^m$ and interval subsets of T. The value of $\bar{g}(x, T_\tau)$ is an interval in $\mathbb{R}$.

In the proposed method, tightened constraints are constructed by using so-called inclusion functions. An interval-valued function $G(x, T_\tau)$ is called an inclusion function of $g(x, y)$ with respect to $y \in T_\tau$, if

$$\bar{g}(x, T_\tau) \subseteq G(x, T_\tau) := [G^l(x, T_\tau), G^u(x, T_\tau)]$$

holds for all interval subsets T_τ of T. $[G^l(x, T_\tau), G^u(x, T_\tau)] \subset \mathbb{R}$ is an interval, which is the value of function $G(\cdot, \cdot)$.

---

[6] Definition of ϵ-optimality: If we denote LB_k and UB_k as sequences generated of upper and lower bounds of the original SIP (5.20), then $\lim_{k \to +\infty} LB_k = \lim_{k \to +\infty} UB_k$ is the minimal objective function value of (5.20). In other words, $\forall \epsilon > 0$, there exist a $k^* > 0$ such that $UB_k - LB_k \leq \epsilon$, $\forall k > k^*$.

https://doi.org/10.51202/9783186952035 - Generiert durch IP 216.73.216.36, am 18.01.2026, 21:11:22. © Urheberrechtlich geschützter Inhalt. Ohne gesonderte
Erlaubnis ist jede urheberrechtliche Nutzung untersagt, insbesondere die Nutzung des Inhalts im Zusammenhang mit, für oder in KI-Systemen, KI-Modellen oder Generativen Sprachmodellen.

To guarantee convergence of the proposed algorithm to ϵ-optimality, one has to make sure additionally that the constructed inclusion function $G(x, T_\tau)$ converges to $\bar{g}(x, T_\tau)$ in a certain sense. Denote

$$\theta([a_l, a_u], [b_l, b_u]) := \max\{|a_l - b_l|, |a_u - b_u|\}$$

as Hausdorf metric of two scalar intervals $[a_l, a_u]$ and $[b_l, b_u]$, a_l, a_u, b_l, $b_u \in \mathbb{R}$. Convergence of function $\bar{g}(x, T_\tau)$ and function $G(x, T_\tau)$ is in the sense that, $\forall T_\tau$, as $m(T_\tau) \to 0$,

$$\begin{aligned} &m(G(x, T_\tau)) \to 0, \\ &\theta(\bar{g}(x, T_\tau), G(x, T_\tau)) \to 0. \end{aligned} \tag{5.36}$$

$m(\cdot)$ is defined as a measure for the size of a n-dimensional interval, e.g., $m(T_\tau) := \max ||y_1 - y_2||$, $\forall y_1, y_2 \in T_\tau$.

A subdivision of T, denoted as $\mathcal{S} := \{T_\tau \mid \tau \in K\}$, satisfies

$$\begin{aligned} &T = \cup_{\tau \in K} T_\tau, \\ &int(T_{\tau_1}) \cap int(T_{\tau_2}) = \emptyset,\ \forall \tau_1 \neq \tau_2, \end{aligned}$$

where K is a finite index set and it relates to the degree of refinement of subdivision $\mathcal{S}$. A subdivision $\mathcal{S}_2$ with index set K_2 is a refinement of a subdivision $\mathcal{S}_1$ with index set K_1, if: (i) $\forall T_{i_2} \in \mathcal{S}_2$ there exist $T_{i_1} \in \mathcal{S}_1$, so that $T_{i_2} \subseteq T_{i_1}$, (ii) there exist $T^*_{i_2} \in \mathcal{S}_2$ and $T^*_{i_1} \in \mathcal{S}_1$ so that $T^*_{i_2} \subset T^*_{i_1}$. That is, a refined subdivision $\mathcal{S}_2$ contains more elements, which are not larger than the elements in the original subdivision $\mathcal{S}_1$. The above relationship between $\mathcal{S}_1$ and $\mathcal{S}_2$ can be denoted by

$$K_1 \prec K_2.$$

Consider a sequence of refinements K_i, $i = 1, \cdots, \pi$, satisfying

$$K_1 \prec \cdots \prec K_\pi,$$

the following property can be obtained by applying Eq. (5.36) [18]:

$$\theta(\bar{g}(x, T), \cup_{\tau \in K_i} G(x, T_\tau)) \to 0, \text{as } i \to +\infty. \tag{5.37}$$

The approximation gets more accurate as the subdivision gets refined. In a limiting sense, the union of the values of the inclusion functions $\cup_{\tau \in K_i} G(x, T_\tau)$ converges to $\bar{g}(x, T)$ in the sense of Hausdorff metric θ. Eq. (5.37) implies for example that

$$\min_{\tau \in K_i} G^l(x, T_\tau) \nearrow \min_{y \in T} g(x, y),\ \text{ as } i \to \infty\ . \tag{5.38}$$

Eqs. (5.37), (5.38) provide a way to estimate the inner optimization problem (under the assumption that function $G(x, T_\tau)$ can be obtained). They will be used to approximate the feasible region of the original SIP from *inside* (cf. problem (5.40)). Because the feasible region gets smaller than the original SIP, upper bounds of the original SIP can be obtained. This property will be integrated into the overall sB&B algorithm, which will be discussed below.

https://doi.org/10.51202/9783186952035 - Generiert durch IP 216.73.216.36, am 18.01.2026, 21:11:22. © Urheberrechtlich geschützter Inhalt. Ohne gesonderte Erlaubnis ist jede urheberrechtliche Nutzung untersagt, insbesondere die Nutzung des Inhalts im Zusammenhang mit, für oder in KI-Systemen, KI-Modellen oder Generativen Sprachmodellen.

In the sB&B framework, branching is done to the space X of variable x. Branching generates a binary tree of subproblems, which are referred as the nodes (leafs) of the binary tree. For example, the root node (corresponding to the original SIP) can be branched into two subnodes by splitting X into two subsets X_1 and X_2 with $int(X_1) \cap int(X_2) = \emptyset$. Each subnode can be branched further in a similar way. The more deeply a node locates in the tree, the smaller the feasible region of variable x. We denote q as an index of each node and X_q as the branched feasible region of the q-th node. Then, each node in the tree refers to the subproblem

$$\begin{aligned} &\min_x f(x) \\ &\text{s.t. } g(x,y) \geq 0, \forall y \in T, \\ &\quad x \in X_q. \end{aligned} \tag{5.39}$$

Note that problem (5.39) is still a SIP, but compared to the original SIP (5.20), it has a smaller feasible region X_q.

For each problem (5.39), inclusion functions are applied in the proposed algorithm, which leads to an upper-bounding problem in the form of

$$\min_x f(x) \tag{5.40a}$$

$$\text{s.t. } G^l(x, T_\tau) \geq 0, \forall \tau \in K_{\phi(q)}, \tag{5.40b}$$

$$x \in X_q. \tag{5.40c}$$

$\phi(q)$ refers to the depth of the q-th node in the tree and $K_{\phi(q)}$ denotes the degree of refinement of set T for the q-th node. The degree of refinement is selected in a way that it is only dependent on the node's depth, i.e., all nodes with the same depth have the same degree of refinement and the more deeply a node locates in the tree, the more intensive the set T is refined. Note that, since $K_{\phi(q)}$ is a finite set, problem (5.40) is a finite optimization problem, which can be solved by local/global NLP algorithms.

Note also that, since the feasible set of problem (5.40) is smaller than the one of problem (5.39) (cf. Eq. (5.38)), solutions of problem (5.40) are valid upper bounds of problem (5.39). Furthermore, from Eq. (5.37) and the selected refinement strategy for $K_{\phi(q)}$ we see that Eq. (5.40b) provides more and more tightened inner estimates of the feasible set of problem (5.39) as the node's depth $\phi(q)$ increases. As a result, the objective values of problem (5.40) converge to the objective function values of problem (5.39) from above as branching goes more deeply in the tree.

In the approach proposed in [18], the lower-bounding problems are generated by discretizing the set T. In the sB&B framework, each subproblem (5.39) for the q-th node is lower-bounded by

$$\begin{aligned} &\min_x f(x) \\ &\text{s.t. } g(x,y) \geq 0, \forall y \in \bar{T}_{\phi(q)}, \\ &\quad x \in X_q, \end{aligned} \tag{5.41}$$

where $\phi(q)$, as it is defined before, denotes the depth of node q. $\bar{T}_{\phi(q)} \subset T$ is a finite set containing discretization points of T. Because $\bar{T}_{\phi(q)}$ is a subset of T, the feasible set of problem (5.41) is larger than the feasible set of problem (5.39). Therefore, the global minima of problem (5.41) provide lower bounds of problem (5.39).

https://doi.org/10.51202/9783186952035 - Generiert durch IP 216.73.216.36, am 18.01.2026, 21:11:22. © Urheberrechtlich geschützter Inhalt. Ohne gesonderte Erlaubnis ist jede urheberrechtliche Nutzung untersagt, insbesondere die Nutzung des Inhalts im Zusammenhang mit, für oder in KI-Systemen, KI-Modellen oder Generativen Sprachmodellen.

To guarantee the convergence of the sB&B framework, $\bar{T}_{\phi(q)}$ is chosen such that

$$m(\bar{T}_{\phi(q)}, T) \to 0, \text{as } \phi(q) \to +\infty.$$

Hence, as branching goes more deeply into the tree, denser discretization is applied. In summary, the feasible set of problem (5.41) approximates the feasible set of problem (5.39) from outside in a limiting sense. Therefore, the generated sequence of optimal solutions of problem (5.41) approaches the solutions of problem (5.39) from below.

A practical difficulty is that problem (5.41) has to be solved to global optimality to guarantee valid lower bounds. This requires considerable computational power in practice. Even worse, the algorithm generates a tree of nodes and each of them corresponds to a lower bounding problem in the form of Eq. (5.41), which means that subproblems have to be solved to global optimality for multiple times.

The sB&B framework results in a binary tree of subproblems in the form of Eq. (5.39). Each subproblem leads to an upper-bounding problem (5.40) and a lower-bounding problem (5.41). Denote u_q and l_q as the obtained upper and lower bounds for the q-th node by solving problems (5.40) and (5.41), respectively. Denote j, $j = 1, 2, \cdots$, as the index for the major iterations of the algorithm. Denote UB_j and LB_j as the overall upper and lower bounds for the original SIP. The overall sB&B algorithm includes nothing more than two additional procedures: A procedure to update UB_j and LB_j by

$$\begin{aligned} UB_{j+1} &= \min(UB_j, u_{q_1^*}, u_{q_2^*}), \\ LB_{j+1} &= \min(LB_j, l_{q_1^*}, l_{q_2^*}), \end{aligned}$$

where q_1^*, q_2^* denote the indices for branched two subnodes of the q-th node; and a procedure to fathom nodes, which are not needed to be branched further. A node q^* can be fathomed at any iteration j^*, if $l_{q^*} \geq UB_{j^*}$. The algorithm terminates at iteration j^*, if $|UB_{j^*} - LB_{j^*}| \leq \epsilon$, where ϵ is a predefined small number for ϵ-optimality.

5.3.3 A robust design method: Normal vector approach

The normal vector approach (NVA) [113, 119] is a robust design method for dynamic systems. In this method, normal vectors of critical manifolds[7] are used to robustly guarantee design properties. Robust design constraints are transformed by using the normal vectors into a set of certain design constraints, which can be treated directly by NLP solvers. Critical manifolds refer to points, at which the behavior of the system changes qualitatively, including bifurcation points, or points at which state variable constraints and/or output constraints are violated. NVA first considered stability manifolds, which resulted in a robustly stable design of dynamic systems. Later, the approach was extended to consider other types of critical manifolds [53, 82, 121]. The NVA is closely related to the local constraint reduction method of SIP [34, 122]. This section first introduces the NVA based on the geometric understanding of the feasible set of a special class of SIP (cf. also [113, 119]). Local convergence of the NVA for the considered class of SIP will be established afterwards in a rigorous way.

---

[7] A manifold is a topological space that locally resembles Euclidean space near each point. A 2-dimensional manifold in $\mathbb{R}^3$ is a hypersurface, while a 1-dimensional manifold in $\mathbb{R}^3$ is a line. Trivially, manifolds can be understand as "boundaries" in higher dimensional spaces.

https://doi.org/10.51202/9783186952035 - Generiert durch IP 216.73.216.36, am 18.01.2026, 21:11:22. © Urheberrechtlich geschützter Inhalt. Ohne gesonderte Erlaubnis ist jede urheberrechtliche Nutzung untersagt, insbesondere die Nutzung des Inhalts im Zusammenhang mit, für oder in KI-Systemen, KI-Modellen oder Generativen Sprachmodellen.

Geometric Interpretation of the NVA

The NVA can be interpreted geometrically (cf. [113, 119]). Let us consider a special class of SIP (5.20), taking the form

$$\min_{x,y} f(x,y) \tag{5.42a}$$

$$\text{s.t. } g(x+t,y) \geq 0, \forall t \in T, \tag{5.42b}$$

where $x \in \mathbb{R}^m$, $t \in \mathbb{R}^m$, $y \in \mathbb{R}^n$, $f(\cdot,\cdot) \in \mathcal{C}^2 : \mathbb{R}^m \times \mathbb{R}^n \to \mathbb{R}$, $g(\cdot,\cdot) \in \mathcal{C}^2 : \mathbb{R}^m \times \mathbb{R}^n \to \mathbb{R}$. $\mathcal{C}^k$, $k = 1, \cdots, \infty$, denotes the set of k times continuous differentiable functions.

$$T = \{t \in \mathbb{R}^m \,|\, t^T t \leq 1\} \tag{5.43}$$

is a fixed compact set. To simplify the discussion, no additional equality/inequality constraints in problem (5.42) are included. Problem (5.42) is denoted as $\mathcal{P}$.

If one denotes $w := x+t \in \mathbb{R}^m$, w can be interpreted as a vector of uncertain parameters. x and t refer to the nominal values and the uncertainty part of w. y can be interpreted as a vector of certain parameters, whose values are not subject to uncertainty. T represents an uncertainty region of a m-dimensional unit ball in the uncertain parameter space of $w \in \mathbb{R}^m$. We note that the uncertainty region T can be used to approximate parametric uncertainties, which appear frequently in engineering applications (cf. [119]). We note also that although the NVA has been applied to treat more complicated SIP, where, e.g., function $g(\cdot)$ in Eq. (5.42b) is not explicitly given, this analysis will be restricted to discuss problem (5.42).

Denote

$$\mathcal{M} := \{(w^T, y^T)^T \in \mathbb{R}^{m+n} \mid g(w,y) = 0\} \tag{5.44}$$

as the so-called critical manifold with respect to Eq. (5.42b). From functional analysis, we know that, if condition

$$\nabla_w g(w,y) \neq 0 \tag{5.45}$$

holds at a point $(\bar{w}^T, \bar{y}^T)^T$, $\mathcal{M}$ is locally a $(m+n-1)$-dimensional manifold in space $\mathbb{R}^{m+n}$. It is a high-dimensional hyperplane, which separates the regions $\{(w^T, y^T)^T \in \mathbb{R}^{m+n} \mid g(w,y) > 0\}$ and $\{(w^T, y^T)^T \in \mathbb{R}^{m+n} \mid g(w,y) < 0\}$ in $\mathbb{R}^{m+n}$. The normal vector of manifold $\mathcal{M}$ is

$$r_m = \nabla_{w,y} g(w,y) \in \mathbb{R}^{m+n},$$

which is a $(m+n)$-dimensional vector. After projecting r_m into the uncertainty parameter space $\mathbb{R}^m$, we can obtain the normalized normal vector

$$r(w,y) = \frac{\nabla_w g(w,y)}{\|\nabla_w g(w,y)\|} \in \mathbb{R}^m. \tag{5.46}$$

$r(w,y)$ refers to the normal vector in the uncertain parameter space, which is used by the NVA to guarantee robustness.

https://doi.org/10.51202/9783186952035 - Generiert durch IP 216.73.216.36, am 18.01.2026, 21:11:22. © Urheberrechtlich geschützter Inhalt. Ohne gesonderte Erlaubnis ist jede urheberrechtliche Nutzung untersagt, insbesondere die Nutzung des Inhalts im Zusammenhang mit, für oder in KI-Systemen, KI-Modellen oder Generativen Sprachmodellen.

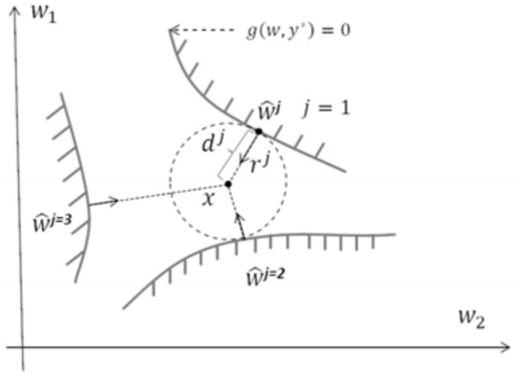


Figure 5.1: A geometric illustration of the normal vector approach. The figure is drawn in the space of uncertain parameters $w \in \mathbb{R}^2$ with fixed certain parameters $y = y^*$. w_1 and w_2 are two elements of vector w. The unit circle refers to the uncertainty region T with x as its center. Solid lines refer to the critical manifold $\mathcal{M}$. $j = 1, 2, 3$ are used to index critical points $\hat{w}^j$. $r^j := r(\hat{w}^j, y^*)$ refers to the normal vector evaluated at critical point $(\hat{w}^{jT}, y^{*T})^T$. d^j represents the closest distance between the center of the uncertainty region and the j-th critical point.

The solution of SIP $\mathcal{P}$ by the NVA can be illustrated by Fig. 5.1. The figure is drawn in the uncertain parameter space by fixing $y = y^*$. For ease of representation, consider that there are only two uncertain parameters, i.e., $w = (w_1, w_2)^T \in \mathbb{R}^2$. Shaded solid lines refer to points satisfying $g(w, y^*) = 0$, which are the projections of the critical manifold $\mathcal{M}$ into the uncertain parameter space. The critical points $\hat{w}^j$ are indexed by j and there exists in total $J = 3$ critical points. The dashed unit circle represents the uncertainty region T and x corresponds to the center of uncertainty region. The robust constraint (5.42b) requires that the unit circle should not overlap with the critical manifold.

The NVA reformulates the uncertainty constraint (5.42b) into a finite number of constraints by using normal vectors $r^j := r(\hat{w}^j, y^*)$. r^j represents a direction, along which the distance d^j between the critical manifold and the center of the uncertainty circle attains its minimum. Therefore by giving lower bounds to the distances d^j, one can make sure that the uncertainty circle does not cross the critical manifold. The NVA solves the following finite optimization problem

$$\min_{x,y,\hat{w},d} f(x, y) \tag{5.47a}$$

$$\text{s.t.} 0 = g(\hat{w}^j, y), j = 1, \cdots, J, \tag{5.47b}$$

$$x = \hat{w}^j + d^j r(\hat{w}^j, y), j = 1, \cdots ., J, \tag{5.47c}$$

$$d^j \geq 1, j = 1, \cdots, J. \tag{5.47d}$$

J refers to the total number of critical points $\hat{w}^j$, which satisfy Eq. (5.47b). $\hat{w} := (\hat{w}^{1T}, \cdots, \hat{w}^{JT})^T$. $r(\hat{w}^j, y)$, $j = 1, \cdots, J$, denote the projected normal vectors, defined in Eq. (5.46). Eq. (5.47c) ensures that, the center of the dashed unit circle x is con-

https://doi.org/10.51202/9783186952035 - Generiert durch IP 216.73.216.36, am 18.01.2026, 21:11:22. © Urheberrechtlich geschützter Inhalt. Ohne gesonderte Erlaubnis ist jede urheberrechtliche Nutzung untersagt, insbesondere die Nutzung des Inhalts im Zusammenhang mit, für oder in KI-Systemen, KI-Modellen oder Generativen Sprachmodellen.

nected with the critical points $\hat{w}^j$, $j = 1, \cdots, J$, along normal vector directions. $d^j \in \mathbb{R}$, $j = 1, \cdots, J$, represent the closest distances between the center of the unit circle and the corresponding critical points. For compact reference, we introduce $d := (d^1, \cdots, d^J)^T$. Eq. (5.47d) guarantees that all these distances should be bigger than 1 for robustness.

In summary, Problem (5.47) actually represents the feasible set of SIP $\mathcal{P}$ by lower bounds on the minimal distances between the center of the unit uncertainty circle to the critical manifold.

On the local convergence of the NVA

To establish the local convergence of the NVA rigorously, we prove here that the feasible set of the SIP $\mathcal{P}$ is locally identical to the feasible set of a derived finite optimization problem (5.68) by using the normal vectors under the conditions that (1) the transversality condition holds at the global minima of the inner NLP, and (2) the second-order sufficient optimality condition of the inner NLP is fulfilled (cf. Theorem 5.3.10). In this sense, local minima of the original infinitely-constrained optimization problem $\mathcal{P}$ are identical to the ones of the derived finitely-constrained optimization problem (5.68), which can be identified by using local NLP algorithms.

The main result is presented in Theorem 5.3.10, which is proved in Appendix E. To introduce the notation, we present some relevant definitions and lemmata. Denote $z = (x^T, y^T)^T \in \mathbb{R}^{m+n}$ as the concatenation of x and y, and denote V_z as a neighborhood of z. Denote the inner optimization problem of $\mathcal{P}$ as $\mathcal{I}(x, y)$. $\mathcal{I}(x, y)$ takes the form

$$\min_{t \in \mathbb{R}^m} g(x + t, y) \quad \text{s. t.} \;\; t^T t \leq 1. \tag{5.48}$$

Denote the Lagrange function of $\mathcal{I}(x, y)$ by

$$L(x, y, t, l) = g(x + t, y) + l(t^T t - 1), \tag{5.49}$$

where $l \in \mathbb{R}$ is the Lagrange multiplier of $\mathcal{I}(x, y)$.

The feasible set of $\mathcal{P}$ is

$$F = \{z = (x^T, y^T)^T \mid g(x + t, y) \geq 0, \forall t \in T\}. \tag{5.50}$$

Define

$$T_a(x, y) = \{t \in T \mid g(x + t, y) = 0\} \tag{5.51}$$

as the active index set of $g(x + \cdot, y) = 0$. $T_a(x, y)$ can also be denoted by using

$$T_a(x, y) = \{t^j \,|\, j \in \mathcal{J}\},$$

where $\mathcal{J} = \{1, \cdots, J\}$ is an index set ($\mathcal{J}$ can be an infinite set and in general the number of elements in $\mathcal{J}$ depend on the evaluation point z). We always assume $T_a(z) \neq \emptyset$. If $T_a(z) = \emptyset$, $\forall z \in F$, problem $\mathcal{P}$ is locally unconstrained.

The following lemma is elementary and we omit the proof.

Lemma 5.3.3. *If $\bar{z} \in F$, $T_a(\bar{z}) \neq \emptyset$, then any $\bar{t} \in T_a(z)$ is a global minimum of the inner problem $\mathcal{I}(\bar{z})$.*

https://doi.org/10.51202/9783186952035 - Generiert durch IP 216.73.216.36, am 18.01.2026, 21:11:22. © Urheberrechtlich geschützter Inhalt. Ohne gesonderte Erlaubnis ist jede urheberrechtliche Nutzung untersagt, insbesondere die Nutzung des Inhalts im Zusammenhang mit, für oder in KI-Systemen, KI-Modellen oder Generativen Sprachmodellen.

Definition 5.3.1 (Transversality condition (TC))**.** *The transversality condition (TC) holds at $z = \bar{z}$, if*

$$\nabla_x g(\bar{x}+t, \bar{y}) \neq 0, \ \forall t \in T_a(\bar{x}, \bar{y}). \tag{5.52}$$

Note that for scalar-valued function $g(\cdot, y) : \mathbb{R}^m \to \mathbb{R}$, $\nabla_x g(x, y)$ is a m-dimensional row vector, while $\nabla_x^T g(x, y)$ is a m-dimensional column vector. $\nabla_{xx} g(x, y)$ is the m-by-m Hessian matrix of $g(\cdot, y)$.

Denote

$$r(x+t, y) = \frac{\nabla_x^T g(x+t, y)}{||\nabla_x g(x+t, y)||}, \tag{5.53}$$

which is a function of x, y and t. r is the so-called normal vector of function $g(x+t, y)$ (cf. Eq. (5.46)). Obviously, under TC, the normal vectors are properly defined for all $t \in T_a(\bar{z})$ and they satisfy $||r|| = 1$. Note that TC is the weakest condition to apply the NVA, since if TC is violated the normal vectors $r(x+t, y)$ are not defined. We note that for vector-valued functions $r(\cdot, y) : \mathbb{R}^m \to \mathbb{R}^m$, $\nabla_x r(x, y)$ is a m-by-m matrix.

An important consequence of TC is that, equation $g'(t) = g(\bar{x}+t, \bar{y}) = 0$ locally defines a $(m-1)$-dimensional hypersurface in $\mathbb{R}^m$ and $g'(t)$ changes its sign across this hypersurface. This is summarized in the following lemma:

Lemma 5.3.4. *Assume that TC holds at $z = \bar{z}$, then $\forall \bar{t} \in T_a(\bar{z})$: (i) $g'(t) = g(\bar{x}+t, \bar{y}) = 0$ locally defines an $(m-1)$-dimensional hypersurface in $\mathbb{R}^m$, (ii) $g'(t)$ changes its sign from negative to positive at $t = \bar{t}$ along the (normal vector) direction $\delta = r(\bar{x}+\bar{t}, \bar{y})$.*

Proof. Because $\nabla_t g'(\bar{t}) = \nabla_x g(\bar{x}+\bar{t}, \bar{y}) \neq 0$, the results of (i) follows from the implicit function theorem (IFT) (cf. Theorem 2.5.1). The results of (ii) can be derived by applying Taylor series to $g'(t)$ along the direction δ. ☐

Lemma 5.3.5. *Assume that $\bar{z} \in F$, $T_a(\bar{z}) \neq \emptyset$ and TC holds at $z = \bar{z}$. Then $\forall \, \bar{t} \in T_a(\bar{z})$, $||\bar{t}|| = 1$.*

Proof. Assume that $\exists \ t' \in T_a(\bar{z})$ with $||t'|| < 1$. From Lemma 5.3.4, direction $-\delta = -r(\bar{x}+t', \bar{y})$ is a decreasing direction of $g'(t) = g(\bar{x}+t, \bar{y})$. Hence, for $\epsilon > 0$ sufficiently small, $t' - \epsilon\,\delta \in T$ and $g(\bar{x}+t'-\epsilon\,\delta, \bar{y}) < g(\bar{x}+t', \bar{y}) = 0$, i.e. $\bar{z} \notin F$, which is a contradiction. ☐

Lemma 5.3.6 (First-order optimality condition of the $\mathcal{I}(\bar{z})$)**.** *Assume that $\bar{z} \in F$, $T_a(\bar{z}) \neq \emptyset$ and that TC holds at $z = \bar{z}$, then $\forall \bar{t} \in T_a(\bar{z})$: (i) Linear independence constraint qualification (LICQ) condition (cf. Definition D.1 in Appendix D) of $\mathcal{I}(\bar{z})$ holds, (ii) strict complementarity (SC) condition (cf. Definition D.2 in Appendix D) of $\mathcal{I}(\bar{z})$ holds, (iii) there exists a unique Lagrange multiplier*

$$\bar{l} = \frac{||\nabla_t g(\bar{x}+\bar{t}, \bar{y})||}{2} > 0, \tag{5.54}$$

such that

$$\nabla_t g(\bar{x}+\bar{t}, \bar{y}) = -2\,\bar{l}\,\bar{t}^T. \tag{5.55}$$

Proof. From Lemma 5.3.3, $\bar{t}$ is a local minimum. From Lemma 5.3.5, LICQ condition of $\mathcal{I}(\bar{z})$ holds. Then there exists a unique Lagrange multiplier $\bar{l} \geq 0$ such that Eq. (5.55) (cf. Theorem D.1 in Appendix D). The value of $\bar{l}$ can be derived by taking norms of both sides of Eq. (5.55). Under TC, we have $\bar{l} > 0$, which indicates the fulfillment of SC condition of $\mathcal{I}(\bar{z})$. ☐

https://doi.org/10.51202/9783186952035 - Generiert durch IP 216.73.216.36, am 18.01.2026, 21:11:22. © Urheberrechtlich geschützter Inhalt. Ohne gesonderte Erlaubnis ist jede urheberrechtliche Nutzung untersagt, insbesondere die Nutzung des Inhalts im Zusammenhang mit, für oder in KI-Systemen, KI-Modellen oder Generativen Sprachmodellen.

Define

$$h(x,y,t,d) = \begin{pmatrix} t + d\, r(x+t,y) \\ g(x+t,y) \end{pmatrix} \tag{5.56}$$

as a vector-valued function, where $d \in \mathbb{R}$. Consider the following equation system

$$h(x,y,t,d) = 0, \tag{5.57}$$

we now prove that under the second-order sufficient conditions (SOSC) of $\mathcal{I}(x,y)$ (cf. Appendix D), Eq. (5.57) locally determines $\mathcal{C}$-functions $t(x,y)$ and $d(x,y)$.

Denote

$$W(x,y,t,l) = \nabla_{tt}L(x,y,t,l) = \nabla_{tt}g(x+t,y) + 2\, l\, I, \tag{5.58}$$

where $I \in \mathbb{R}^{m\times m}$ is an identity matrix.

Definition 5.3.2 (Second-order sufficient conditions (SOSC) of $\mathcal{I}(x,y)$, cf. Theorem D.2 in Appendix D)**.** *The second-order sufficient condition (SOSC) of* $\mathcal{I}(\bar z)$ *holds at* $\bar t$, *if* $(\bar t^T, \bar l)^T$ *is a KKT point of* $\mathcal{I}(\bar z)$*, namely they satisfy Eq. (5.55),* $\bar t^T \bar t \le 1$ *and* $\bar l \ge 0$*, and if*

$$s^T W(\bar x, \bar y, \bar t, \bar l) s > 0,\ \forall s \in \{s \in \mathbb{R}^m \,|\, \bar t^T s = 0, s \neq 0\}. \tag{5.59}$$

Lemma 5.3.7. *Assume that* $\bar z \in F$, $T_a(\bar z) \neq \emptyset$, *TC holds at* $z = \bar z$ *and SOSC (5.59) of* $\mathcal{I}(\bar z)$ *holds, then: (i) any* $\bar t \in T_a(\bar z)$ *is a locally isolated (locally unique) local minimizer of* $\mathcal{I}(\bar z)$*, (ii)* $\forall \bar t \in T_a(\bar z)$*, the local minimum of* $\mathcal{I}(z)$ *can be locally described by a* $\mathcal{C}$*-function* $t(x,y)$ *for* t *near* $\bar t$*, (iii)* $T_a(\bar z)$ *is a finite set.*

Proof. From Lemma 5.3.6, LICQ and SC of $\mathcal{I}(z)$ hold. The proofs of (i) and (ii) follow directly from Theorem D.2 and D.3 in Appendix D. To prove (ii), because T is a compact set and $T_a(\bar z) \subseteq T$ contains isolated points, $T_a(\bar z)$ must be a finite set. □

Lemma 5.3.8. *Assume* $\bar z \in F$, $T_a(\bar z) \neq \emptyset$ *and TC holds at* $z = \bar z$. $\forall \bar t \in T_a(\bar z)$*, we have*

$$h(\bar x, \bar y, \bar t, \bar d) = 0, \tag{5.60}$$

with $\bar d = 1$.

Proof. From the definition of $T_a(\bar z)$, it is obvious that $g(\bar x + \bar t, \bar y) = 0$, $\forall \bar t \in T_a(\bar z)$. Now we prove that

$$\bar t + r(\bar x + \bar t, \bar y) = 0. \tag{5.61}$$

From Lemma 5.3.6, replacing $\bar l$ in Eq. (5.55) by Eq. (5.54) leads to

$$\nabla_t g(\bar x + \bar t, \bar y) + ||\nabla_t g(\bar x + \bar t, \bar y)||\, \bar t^T = 0.$$

Eq. (5.61) follows directly by using the property $\nabla_t g(\bar x + \bar t, \bar y) = \nabla_x g(\bar x + \bar t, \bar y)$. □

Lemma 5.3.9. *Assume* $\bar z \in F$, $T_a(\bar z) \neq \emptyset$ *and TC holds at* $z = \bar z$. *If the SOSC of* $\mathcal{I}(\bar z)$*, namely Eq. (5.59), is fulfilled* $\forall \bar t \in T_a(\bar z)$*, then the Jacobian matrix of* $h(x,y,t,l)$ *(with respect to* t*,* d*) is non-singular at* $x = \bar x$*,* $y = \bar y$*,* $t = \bar t$ *and* $d = \bar d = 1$*, i.e.,*

$$|\nabla_{t,d} h(\bar x, \bar y, \bar t, \bar d)| \neq 0. \tag{5.62}$$

https://doi.org/10.51202/9783186952035 - Generiert durch IP 216.73.216.36, am 18.01.2026, 21:11:22. © Urheberrechtlich geschützter Inhalt. Ohne gesonderte Erlaubnis ist jede urheberrechtliche Nutzung untersagt, insbesondere die Nutzung des Inhalts im Zusammenhang mit, für oder in KI-Systemen, KI-Modellen oder Generativen Sprachmodellen.

Proof. It is straightforward to see that

$$\begin{aligned}\nabla_t r(x+t,y) &= \nabla_t\left(\frac{\nabla_t^T g(x+t,y)}{\|\nabla_t g(x+t,y)\|}\right)\\ &= \nabla_t^T g(x+t,y)\,\nabla_t\left(\frac{1}{\|\nabla_t g(x+t,y)\|}\right)+\frac{\nabla_{tt} g(x+t,y)}{\|\nabla_t g(x+t,y)\|}\\ &= \frac{-\nabla_t^T g\nabla_t g\nabla_{tt} g}{\|\nabla_t g\|^3}+\frac{\nabla_{tt} g}{\|\nabla_t g\|},\end{aligned}\tag{5.63}$$

and

$$\nabla_{t,d} h(\bar{x},\bar{y},\bar{t},\bar{d}) = \begin{pmatrix} I+\nabla_t r(\bar{x}+\bar{t},y) & r(\bar{x}+\bar{t},\bar{y}) \\ \nabla_t g(\bar{x}+\bar{t},\bar{y}) & 0 \end{pmatrix} := H.$$

Denote $v = (s_1^T, s_2)^T$ with $s_1 \in \mathbb{R}^m$, $s_2 \in \mathbb{R}$. To prove Eq. (5.62), we need to prove that $v = 0$ is the *unique* solution of $H\,v = 0$, i.e.

$$(I+\nabla_t r)s_1 + r s_2 = 0, \tag{5.64a}$$

$$\nabla_t g\, s_1 = 0, \tag{5.64b}$$

have a unique solution of $s_1 = 0$, $s_2 = 0$. Assume that there exists a non-zero solution $v^* = (s_1^*, s_2^*)$ with $s_1^* = 0$. Because $r(\bar{x}+\bar{t},\bar{y}) \neq 0$ from TC, Eq. (5.64a) results in $s_2^* = 0$, which is a contradiction. Assume now that there exists a non-zero solution $v^* = (s_1^*, s_2^*)$ with $s_1^* \neq 0$. From Eq. (5.64b) and Eqs. (5.54), (5.55), we have

$$\bar{t}^T s_1^* = 0. \tag{5.65}$$

Multiplying Eq. (5.64a) by s_1^{*T} from the left side, we have

$$\begin{aligned} 0 &= s_1^{*T}(I+\nabla_t r)s_1^*\\ &= s_1^{*T}(I+\frac{\nabla_{tt} g}{\|\nabla_t g\|})s_1^*\\ &= s_1^{*T}(I+\frac{\nabla_{tt} g}{2\bar{l}})s_1^*\\ &= \frac{1}{2\bar{l}}s_1^{*T}W(\bar{x},\bar{y},\bar{t},\bar{l})s_1^*,\end{aligned}\tag{5.66}$$

where the first equality holds because $s_1^{*T} r = 0$ as it is required in Eq. (5.64b). The second equality holds because of Eq. (5.63) and Eq. (5.64b). The third equality holds because of Eqs. (5.54). Eqs. (5.65), (5.66) are contradictory to the SOSC (5.59) of $\mathcal{I}(\bar{x},\bar{y})$. □

From the previous lemma and if we denote

$$T_a(\bar{z}) = \{t^1, \cdots, t^J\},\ J < \infty,$$

as a finite set (cf. Lemma 5.3.7), Eq. (5.60) locally determines J $\mathcal{C}$-functions $t^j(x,y)$, $d^j(x,y)$, satisfying $t^j(\bar{x},\bar{y}) = t^j$, $d^j(\bar{x},\bar{y}) = 1$, $\forall\, j = 1,\cdots,J$. Denote

$$F^n = \{z = (x^T, y^T)^T \mid d^j(x,y) \geq 1, j \in 1,\cdots,J\}. \tag{5.67}$$

Our main result is the following theorem:

https://doi.org/10.51202/9783186952035 - Generiert durch IP 216.73.216.36, am 18.01.2026, 21:11:22. © Urheberrechtlich geschützter Inhalt. Ohne gesonderte Erlaubnis ist jede urheberrechtliche Nutzung untersagt, insbesondere die Nutzung des Inhalts im Zusammenhang mit, für oder in KI-Systemen, KI-Modellen oder Generativen Sprachmodellen.

Theorem 5.3.10 (Normal vector reduction theorem)**.** *Assume $\bar{z} \in F$, $T_a(\bar{z}) \neq \emptyset$, TC holds at $z = \bar{z}$ and the SOSC (5.59) of $\mathcal{I}(\bar{x}, \bar{y})$ is fulfilled for all $\bar{t} \in T_a(\bar{z})$. Then, there exists a neighborhood $V_{\bar{z}}$ of $\bar{z}$, such that*

$$V_{\bar{z}} \cap F = V_{\bar{z}} \cap F^n.$$

Proof. This theorem follows as a direct consequence of Theorem E.1.6 and Theorem E.2.5, proved in Appendix E. □

Therefore, to locally identify the local minima of SIP $\mathcal{P}$, one can solve $\mathcal{P}^n$ defined by

$$\begin{aligned} &\min_{x,y} f(x, y) \\ &\text{s. t. } d^j(x, y) \geq 1, \ \forall j = 1, \cdots, J, \end{aligned} \tag{5.68}$$

where $d^j(x, y)$ are implicitly defined by Eq. (5.56). If $\bar{z}$ is a local minimum of the original SIP $\mathcal{P}$, it is also a local minimum of NLP $\mathcal{P}^n$, and vice versa. This way, we reduce an infinitely-constrained problem $\mathcal{P}$ to a finitely-constrained optimization problem $\mathcal{P}^n$. Local convergence to optimal solution $\bar{z}$ of $\mathcal{P}$ can be therefore guaranteed by solving NLP $\mathcal{P}^n$, if the initialization point of $\mathcal{P}^n$ is sufficiently close to $\bar{z}$ and proper convergence conditions of NLP $\mathcal{P}^n$ are fulfilled at $\bar{z}$ in addition. For example, if the second-order sufficient condition of NLP $\mathcal{P}^n$ is filled at $\bar{z}$, one can apply quasi-Newton iterations to identify $\bar{z}$ by solving the KKT system of $\mathcal{P}^n$.

Problem $\mathcal{P}^n$ is closely related to the original presentation (5.47) presented by Mönnigmann and Marquardt [119]. $d^j(x, y)$ refer to the distances from a feasible point $z = (x^T, y^T)^T$ to the critical point $\hat{w}^j$ on the critical manifold $\mathcal{M}$, defined in Eq. (5.44). It is implicitly defined by the equation system (5.57) corresponding to Eqs. (5.47b), (5.47c). For $\mathcal{P}^n$, J refers to the total number of elements in set $T_a(x, y)$, while for problem (5.47) it refers to the total number of critical points (cf. Fig. 5.1).

We note at the end that the obtained results in Theorem 5.3.10 refer to local convergence of the NVA, because the feasible set is only *locally* identical and we do not know how large the neighborhood $V_{\bar{z}}$ can be. The value of J depends also on the evaluation point $\bar{z}$. In order to identify the local minimum $\bar{z}$ of $\mathcal{P}$ by solving $\mathcal{P}^n$, the initial guess of z must be sufficiently close to $\bar{z}$. If remote initial points are selected, the solution of $\mathcal{P}^n$ may not converge to the local minimium of $\mathcal{P}$. Note also that to ensure the global convergence property, Mönnigmann and Marquardt [119] have proposed an iterative procedure to identify the local minima of the original SIP from a remote starting point by detecting critical manifolds. To the author's knowledge, however, there does not exist a rigorous convergence proof. Establishing the global convergence of the NVA is out of the scope of this work and it remains an interesting question for the future.

5.4 Eigenvalue optimization

Optimization problems containing eigenvalue functions (2.5) in the general form are called eigenvalue optimization (EVO) problems. In this section, we consider the following type of EVO problems

$$\begin{aligned} &\min_{x} f(x) \\ &s.t. \ \alpha_{M(x)}(x) \leq -c, \end{aligned} \tag{5.69}$$

https://doi.org/10.51202/9783186952035 - Generiert durch IP 216.73.216.36, am 18.01.2026, 21:11:22. © Urheberrechtlich geschützter Inhalt. Ohne gesonderte
Erlaubnis ist jede urheberrechtliche Nutzung untersagt, insbesondere die Nutzung des Inhalts im Zusammenhang mit, für oder in KI-Systemen, KI-Modellen oder Generativen Sprachmodellen.

where $x \in \mathbb{R}^m$ and $M(x) \in \mathcal{M}_n$. Function f is sufficiently smooth. $\alpha_{M(x)}(x)$, defined in Eq. (2.7), denotes the spectral abscissa of matrix $M(x)$. $c \geq 0$ is a given constant. Other nonlinear equality and inequality constraints are may be present but not explicitly shown in problem (5.69) for simplicity.

We note that since smooth optimization techniques (both, local and global algorithms) are based on the assumption that both the objective function and all constraints are sufficiently smooth, non-Lipschitz continuity of $\alpha_{M(x)}(x)$ (cf. Section 2.2) prevents the use of standard smooth optimization techniques to solve problem (5.69).

5.4.1 Relation to semi-definite programming

Let $M_s : \mathbb{R}^m \to \mathcal{S}_n$ be a smooth function, where $\mathcal{S}_n$ denotes the vector space of all real symmetric $n \times n$ matrices. Matrix $M_s(x)$, $x \in \mathbb{R}^m$, is called positive semi-definite, if

$$w^T M_s(x) w \geq 0, \ \forall w \in \mathbb{R}^n.$$

A positive semi-definite matrix $M_s(x)$ is denoted as

$$M_s(x) \succeq 0. \tag{5.70}$$

Note that symmetric real matrices have only real eigenvalues and that a symmetric real matrix is positive semi-definite, iff all of its (real) eigenvalues are non-negative. The definition of positive definite, negative semi-definite and negative definite can be introduced in a similar way. For example, a symmetric real matrix $M_s(x) \in \mathcal{S}_n$ is called positive definite, if

$$w^T M_s(x) w > 0, \ \forall w \in \mathbb{R}^n, \ w \neq 0. \tag{5.71}$$

A positive definite matrix $M_s(x)$ can be denote as

$$M_s(x) \succ 0. \tag{5.72}$$

Nonlinear semi-definite programing (SDP) is an optimization problem, which considers Eq. (5.70) as one of its constraints. Linear SDP, however, assumes that $M_s(x)$ is a linear function of x.

SDP with constraint (5.70) is a special case of an EVO problem (5.69), because Eq. (5.70) can be written equivalently as

$$\alpha_{-M_s(x)}(x) \leq 0.$$

Hence, existing methods for eigenvalue optimization problems can be directly applied to SDP.

We refer to [172] and [185] for good reviews of linear and nonlinear SDP. The theory and solution methods for linear SDP are already well developed, but the study of nonlinear SDP is much more recent [185].

Transform EVO into SDP

We next review a method to transform the EVO problem (5.69) into a SDP. The transformation is based on Lyapunov equations [84] and the motivation of the transformation is based on the assumption that SDP can be solved more easily than EVO problems. The transformation is, however, limited to a subclass of EVO problems (5.69). The general treatment of EVO problems (5.69) will be discussed in Section 5.4.2 and 5.4.3.

https://doi.org/10.51202/9783186952035 - Generiert durch IP 216.73.216.36, am 18.01.2026, 21:11:22. © Urheberrechtlich geschützter Inhalt. Ohne gesonderte
Erlaubnis ist jede urheberrechtliche Nutzung untersagt, insbesondere die Nutzung des Inhalts im Zusammenhang mit, für oder in KI-Systemen, KI-Modellen oder Generativen Sprachmodellen.

The subclass of problem (5.69) considered is generated by setting c in Eq. (5.69) to a very small constant, e.g., $1.0E{-}6$. The eigenvalue constraint can be then approximately written as

$$\alpha_{M(x)}(x) < 0. \tag{5.73}$$

Optimization problems with constraint (5.73) appear often as stability problems in engineering applications. In this sense, $M(x)$ represents the Jacobian matrix of a nonlinear dynamic system (cf. Section 2.3 and 2.4).

The following theorem introduces a link between the eigenvalue constraint (5.73) and the semi-definite constraint (5.70).

Theorem 5.4.1 (Lyapunov equations, Theorem 3.6 in [84])**.** *For any (non-symmetric) real matrix $X \in \mathcal{M}_n$, $\alpha_X < 0$, iff for any given positive definite symmetric matrix $Q \in \mathcal{S}_n$ there exist a positive definite symmetric matrix $P \in \mathcal{S}_n$ that satisfies*

$$PX + X^T P + Q = 0. \tag{5.74}$$

More over, if $\alpha_X < 0$, then P is a unique solution of Eq. (5.74).

Eq. (5.74) is called a Lyapunov equation. The above theorem says that, $\forall Q \succ 0$, Eq. (5.73) is equivalent to the feasibility of the following equation system

$$PM(x) + M(x)^T P + Q = 0, \tag{5.75a}$$

$$P \succ 0. \tag{5.75b}$$

In practice, one can choose Q as the identity matrix $I \in \mathcal{S}_n$. Hence, using the Lyapunov equation (5.75), eigenvalue constraint (5.73) involving a non-symmetric matrix $M(x)$ can be transformed to a set of nonlinear equalities and a positive definite constraint involving a symmetric matrix P. To obtain a SDP (with semi-definite inequalities), one can approximately replace Eq. (5.75b) by

$$P \succeq \epsilon I,$$

where $\epsilon > 0$ is a small constant. Blanco and Bandoni [21] have first applied this SDP-based reformulation to solve eigenvalue optimization problems with constraint (5.73).

5.4.2 Solution methods for EVO: Non-smooth optimization

Having introduced an SDP-based reformulation to treat a subclass of the EVO (5.69), we next review solution methods for EVO (5.69). We classify the methods into non-smooth optimization techniques (Section 5.4.2) and smoothing techniques (Section 5.4.3). Non-smooth optimization attempts to treat the non-smoothness of eigenvalue constraints directly, while smoothing techniques try to approximate the original non-smooth eigenvalue constraints by smoothing. From the application point of view and with respect to robustness of the solution methods, smoothing techniques seem to be more favorable, because one can use off-the-shelf NLP solvers, which are efficient and reliable.

Non-smooth optimization (NSO) refers to an optimization problem, in which the objective function and/or the constraints are typically not differentiable. Because of the non-differentiability, classical techniques developed for smooth optimizations may fail to find local optimal points of NSO.

https://doi.org/10.51202/9783186952035 - Generiert durch IP 216.73.216.36, am 18.01.2026, 21:11:22. © Urheberrechtlich geschützter Inhalt. Ohne gesonderte
Erlaubnis ist jede urheberrechtliche Nutzung untersagt, insbesondere die Nutzung des Inhalts im Zusammenhang mit, für oder in KI-Systemen, KI-Modellen oder Generativen Sprachmodellen.

The state-of-the-art optimization algorithms for NSO are mainly restricted to unconstrained NSO in the form of

$$\min_{x} f^n(x), \tag{5.76}$$

where $x \in \mathbb{R}^m$ and f^n is locally Lipschitz continuous [8]. When a constrained NSO is supposed to be solved, one can utilize exact penalty functions to transform the constrained into an unconstrained NSO (cf. Chapter 16 in [8]). Consider the constrained NSO

$$\begin{aligned} &\min_{x} f^n(x) \\ s.t.\ &h_i^n(x) = 0, i = 1, \cdots, p, \\ &g_j^n(x) \leq 0, j = 1, \cdots, q. \end{aligned} \tag{5.77}$$

f^n, h_i^n, and g_j^n are locally Lipschitz continuous functions. The l_1 exact penalty function for problem (5.77) is defined by

$$P_r(x) = f^n(x) + r\left(\sum_{i=1,\cdots,p} |h_i^n(x)| + \sum_{j=1,\cdots,q} max\{0, g_j^n(x)\}\right).$$

An important feature of the exact penalty function $P_r(x)$ is that, if $r > 0$ is large enough, local minimizers of $P_r(x)$ are exactly the same as the local minimizers of the constrained problem (5.77). Hence, using exact penalty functions allows us to only consider solution methods for the unconstrained NSO (5.76).

We have to stress that the current status of NSO is restricted to Lipschitz continuous functions [8]. Therefore the EVO problem (5.69), in which the spectral abscissa function (2.7) is non-Lipschitz continuous, can not be directly treated. There exist, however, some successful attempts [25, 27], which solve EVO by using the methods developed for NSO. But to the author's knowledge, a convergence proof is still missing.

Subgradient and an optimality condition

We briefly review next the fundamentals of subgradients and optimality conditions for NSO [8]. Like in smooth optimization, NSO uses generalized forms of "gradients", i.e., the subgradients and the generalized subgradients, to characterize local minima of NSO.

Definition 5.4.1 (Subdifferential and subgradient)**.** *The subdifferential of a convex function $f : \mathbb{R}^m \to \mathbb{R}$ is the set $\partial_c f$ of vectors $v \in \mathbb{R}^m$ such that*

$$\partial_c f(x) = \{v \in \mathbb{R}^m | f(y) \geq f(x) + v^T(y - x),\ \forall y \in \mathbb{R}^m\}. \tag{5.78}$$

Each vector $v \in \partial_c f(x)$ is called a subgradient of f at x.

Definition 5.4.2 (Generalized subdifferential and subgradient)**.** *Let $f : \mathbb{R}^m \to \mathbb{R}$ be locally Lipschitz continuous at point $x \in \mathbb{R}^m$, then the generalized subdifferential of f at x is the set ∂f of vectors $v \in \mathbb{R}^n$ such that*

$$\partial f = \{v \in \mathbb{R}^m | \underbrace{\limsup_{y \to x, t \searrow 0} \frac{f(y + td) - f(y)}{t}}_{:= f^o(x,d)} \geq v^T d,\ \forall d \in \mathbb{R}^m\}.$$

Each vector $v \in \partial f(x)$ is called a generalized subgradient of f at x.

https://doi.org/10.51202/9783186952035 - Generiert durch IP 216.73.216.36, am 18.01.2026, 21:11:22. © Urheberrechtlich geschützter Inhalt. Ohne gesonderte Erlaubnis ist jede urheberrechtliche Nutzung untersagt, insbesondere die Nutzung des Inhalts im Zusammenhang mit, für oder in KI-Systemen, KI-Modellen oder Generativen Sprachmodellen.

$f^o(x, d)$ is the so-called generalized directional derivative of $f(\cdot)$ at x in direction d.

The generalized subdifferential is a generalization of the classical gradient of smooth non-convex functions. Hence, if $f(\cdot)$ is continuously differentiable at x, then $\partial f = \{\nabla f\}$. Furthermore, the generalized subdifferential degenerates to the subdifferential, if the considered function $f(\cdot)$ is convex. If $f(\cdot)$ is a convex function, then

$$\partial_c f = \partial f.$$

Note that, because every convex function is locally Lipschitz, both subgradients and generalized subgradients are defined for Lipschitz continuous functions. Therefore subgradients and generalized subgradients can not be applied to the eigenvalue function (2.7), which is non-Lipschitz continuous.

Now let us return to NSO (5.76). Generalized subdifferentials will lead to a direct way to characterize the local minima of NSO.

Theorem 5.4.2 (A necessary optimality condition for NSO [8])**.** *Assume that $f^n(\cdot)$ in NSO (5.76) attains a local minimum at x^*, and it is locally Lipschitz continuous at x^*, then*

$$\begin{aligned} &(1)\ 0 \in \partial f^n(x^*), \\ &(2)\ f^o(x^*, d) \geq 0,\ \forall d \in \mathbb{R}^m. \end{aligned} \tag{5.79}$$

From this theorem, we can see that, finding point x^* that satisfies condition (5.79) is different from finding solutions of the KKT conditions of NLP. This is because $\partial f^n(x^*)$ is in general a non-finite set. The full description of $\partial f^n(x^*)$ can not easily be obtained during numerical iterations. Typically, only a random element, i.e., a generalized subgradient, belonging to set $\partial f^n(x^*)$ can be numerically calculated [8]. Therefore, the set $\partial f^n(x^*)$ is not completely known. This is a major difference between algorithms for NSO and NLP.

A solution algorithm: the bundle method

Solution algorithms of NSO include subgradient methods, cutting plane methods, bundle methods and gradient sampling methods. All of them are local methods, i.e., they do not attempt to find the global minimum. These algorithms are based on the assumption that only the objective function value and an arbitrary (generalized) subgradient are available at each iteration point. For a comprehensive review of these algorithms, we refer to [8]. Here we outline the major features of the bundle methods only.

Bundle methods are regarded as the most effective and reliable methods for NSO [8]. The basic idea of bundle methods is to approximate the subdifferential of the objective function by gathering subgradients from previous iterations. More information about the local behavior of the objective function can be obtained, compared to the case where an arbitrary subgradient is evaluated at each iteration.

Denote x_k, $k = 1, 2, \cdots$, as iteration points. Denote y_j, $j = 1, 2, \cdots$, as the points from the past iterations, where a subgradient $v_j \in \partial f(y_j)$ is already evaluated. J_k is a nonempty index set of $\{1, \cdots, k\}$. Denote

$$\hat{f}_k^n(x) = \max_{j \in J_k} \{f^n(y_j) + v_j^T(x - y_j)\}$$

as a linear function, which approximates the objective function $f^n(x)$.

https://doi.org/10.51202/9783186952035 - Generiert durch IP 216.73.216.36, am 18.01.2026, 21:11:22. © Urheberrechtlich geschützter Inhalt. Ohne gesonderte Erlaubnis ist jede urheberrechtliche Nutzung untersagt, insbesondere die Nutzung des Inhalts im Zusammenhang mit, für oder in KI-Systemen, KI-Modellen oder Generativen Sprachmodellen.

At each iteration k, bundle methods, more exactly proximal bundle methods and bundle trust methods, compute a descent direction d_k and step sizes t_k^1 and t_k^2 such that

$$\begin{aligned} x_{k+1} &= x_k + t_k^1 d_k, \\ y_{k+1} &= x_k + t_k^2 d_k. \end{aligned}$$

The descent direction d_k in proximal bundle methods is computed by

$$d_k \in \arg \min_{d \in \mathbb{R}^m} \hat{f}_k^n(x_k + d) + \frac{1}{2} u_k d^T d,$$

where $u_k > 0$ is a properly selected weighting parameter, which guarantees the existence of a solution d_k. The descent direction d_k in bundle trust methods is computed by

$$d_k \in \arg \min_{d \in \mathbb{R}^m} \hat{f}_k^n(x_k + d), \ s.t. \ d^T d \leq \sigma_k,$$

where $\sigma_k > 0$ denotes the radius of the trust region. For properly selected step sizes t_k^1 and t_k^2, the proximal bundle methods and the bundle trust methods are proven to be globally convergent [8].

Trials to solve eigenvalue problems

Because the SA function in Eq. (2.7) violates the assumption of Lipschitz continuity, upon which the above-mentioned NSO techniques are based, their direct application to solve EVO (5.69) is problematic. However, there exist some trials [25, 27] which utilize NSO techniques to solve EVO.

A random gradient bundle method, which is inspired by the gradient bundle method, is proposed to solve matrix stability problems [25]. Burke et al. [27] proposed also a gradient sampling algorithm, which is applied to solve some eigenvalue problems. The authors claimed that the developed methods behave robustly in solving eigenvalue problems.

However, to the author's knowledge, a convergence proof for EVO with non-Lipschitz continuous constraints is still missing. The proposed algorithms in [25, 27] are only proven to be convergent for Lipschitz-continuous functions. To summarize, although there exist some successfully trials in solving EVO by using state-of-the-art NSO techniques, solving EVO by non-smooth techniques still needs further theoretical developments.

5.4.3 Solution methods for EVO: Smoothing techniques

In contrast to the non-smooth optimization techniques, smoothing techniques try to smoothen the eigenvalue function (2.7) and derive NLP problems which approximate the original EVO. Instead of solving the non-smooth EVO, one solves the derived smooth NLP, whose solution approximates the solution of the original EVO. In this section, we review three different smoothing techniques. The smoothing method based on H_2-type function [174] is used to solve the presented case studies later.

A method based on pseudospectral abscissa

The ϵ-pseudospectrum of matrix $M(x) \in \mathcal{M}_n$, denoted as $\Lambda_{M(x)}^{\epsilon}$, is a subset of the complex plane. This subset contains all eigenvalues of complex matrices, which are within a distance ϵ to matrix $M(x)$ in the metric space $\mathcal{M}_n$ [26]. Hence,

$$\Lambda_{M(x)}^{\epsilon} = \{\lambda \in \mathbb{C} \mid \lambda \in \Lambda_X, \text{ where } ||X - M(x)||_2 \leq \epsilon, \ X \in \mathcal{M}_n\},$$

https://doi.org/10.51202/9783186952035 - Generiert durch IP 216.73.216.36, am 18.01.2026, 21:11:22. © Urheberrechtlich geschützter Inhalt. Ohne gesonderte Erlaubnis ist jede urheberrechtliche Nutzung untersagt, insbesondere die Nutzung des Inhalts im Zusammenhang mit, für oder in KI-Systemen, KI-Modellen oder Generativen Sprachmodellen.

where Λ_X denotes the eigenvalue spectrum of X.

The ϵ-pseudospectral abscissa of $M(x)$, denoted as $\alpha^{\epsilon}_{M(x)}(x)$, is defined as the maximal real part of the elements in $\Lambda^{\epsilon}_{M(x)}$. Hence,

$$\alpha^{\epsilon}_{M(x)}(x) = sup\{Re(\lambda) \mid \lambda \in \Lambda^{\epsilon}_{M(x)}\}.$$

$\alpha^{\epsilon}_{M(x)}(x)$ is sometimes called the "robust" spectral abscissa of $M(x)$, because for any x,

$$\alpha^{\epsilon}_{M(x)}(x) > \alpha_{M(x)(x)}, \forall \epsilon > 0, \tag{5.80}$$

$$\lim_{\epsilon \searrow 0} \alpha^{\epsilon}_{M(x)}(x) = \alpha_{M(x)}. \tag{5.81}$$

Therefore, if $\alpha^{\epsilon}_{M(x)}(x) \leq -c$ holds at point x for any $\epsilon > 0$, it is guaranteed that

$$\alpha_{M(x)}(x) \leq -c$$

holds robustly in a neighborhood U of x.

One of the important properties of pseudospectral abscissa is that it is Lipschitz-continuous under mild conditions [26]. Hence, by using pseudospectral abscissa we can smoothen the spectral abscissa function from a non-Lipschitz-continuous function to a Lipschitz-continuous function. Hence, we can apply the existing techniques of non-smooth optimization, reviewed in Section 5.4.2 to approximately solve EVO (5.69)[8]. For example, Burke et al. [25] have used a non-smoothing optimization technique (gradient bundle method) to minimize the pseudospectral abscissa.

Definition 5.4.3 (Geometric multiplicity). *Denote $\lambda_{i^*}(x)$ as an eigenvalue of matrix $M(x) \in \mathcal{M}_n$, geometric multiplicity is the dimension of the following vector space*

$$\{v \in \mathbb{C}^n \mid (M(x) - \lambda_{i^*}(x)I)v = 0\}. \tag{5.82}$$

Definition 5.4.4 (Non-derogatory eigenvalues). *An eigenvalue of matrix $M(x) \in \mathcal{M}_n$ is non-derogatory, if it has geometric multiplicity of one.*

Lipschitz continuity of pseudospectral abscissa can be derived from the following theorem.

Theorem 5.4.3 (Lipschitz continuity of pseudospectral abscissa, Corollary 8.3 in [26]). *If all active eigenvalues of a matrix $M(x) \in \mathcal{M}_n$ are non-derogatory, then for all small $\epsilon > 0$, the pseudospectral abscissa $\alpha^{\epsilon}_{M(x)}(x)$ is locally Lipschitz continuous in a neighborhood of x.*

The pseudospectral abscissa of matrix $M_0(x)$ in Example 2.1 is calculated by using the codes provided by Kressner and Vandereycken [91]. The results are presented in Fig. 5.2. As it can be seen from the figure, the pseudospectral abscissa of $M_0(x)$ is at least Lipschitz continuous. Note that, at the non-smooth point $x = 2$, two eigenvalues are the same, which are both active and non-derogatory.

---

[8] Note that current non-smooth optimization techniques are limited to Lipschitz-continuous functions.

https://doi.org/10.51202/9783186952035 - Generiert durch IP 216.73.216.36, am 18.01.2026, 21:11:22. © Urheberrechtlich geschützter Inhalt. Ohne gesonderte Erlaubnis ist jede urheberrechtliche Nutzung untersagt, insbesondere die Nutzung des Inhalts im Zusammenhang mit, für oder in KI-Systemen, KI-Modellen oder Generativen Sprachmodellen.

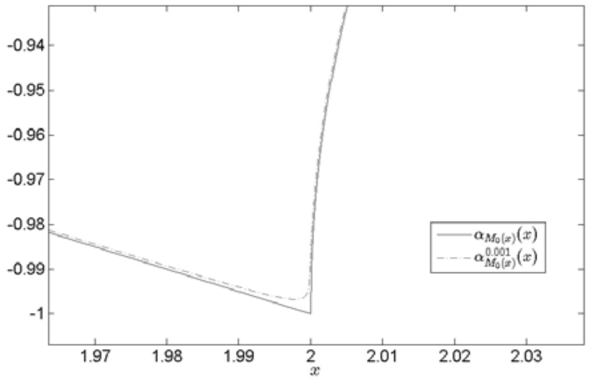


Figure 5.2: Pseudospectral abscissa with $\epsilon = 0.001$ and spectral abscissa of $M_0(x)$.

A method based on H_2-type cost function

Here we present a method based on a relaxed H_2-type cost function to obtain a smoothened function of the spectral abscissa function [174]. A important advantage of this method is that the smoothened function has at least first-order continuous derivatives such existing smooth optimization techniques can be directly applied. To the author's experience, the smoothing method is efficient and reliable, when treating matrices of moderate size.

$\forall X \in \mathcal{M}_n$ and $s \in \mathbb{R}$, define

$$\phi(X, s) = \int_0^\infty ||e^{(X-sI)t}||_F^2 dt, \tag{5.83}$$

where F denotes the Frobenius norm of matrices, i.e. $||X||_F^2 = trace(X^T X)$. $I \in \mathcal{M}_n$ denotes an identify matrix. $\phi(X, s)$ is a real scalar-valued function. The name H_2-type cost function comes from the fact that $\phi(X, s)$ refers to a square-weighted H_2 norm of a dynamic system with transfer function $H_s(z) = (zI - (A - sI))^{-1}$.

It is proven in [174], that if $s > \alpha_{X^*}$, the range of function $\phi(X^*, s)$ takes all positive real numbers. That is, $\forall X^* \in \mathcal{M}_n$,

$$\{\phi(X^*, s) \mid s > \alpha_{X^*}\} = \mathbb{R}_+/\{0\}. \tag{5.84}$$

Furthermore, if $s > \alpha_{X^*}$,

$$\frac{\partial \phi(X^*, s)}{\partial s} < 0. \tag{5.85}$$

According to Eqs. (5.84), (5.85), $f(X^*, s)$ is a monotonic decreasing function of s in the region $\{s \in \mathbb{R} \mid s > \alpha_{X^*}\}$. From the fact that $\phi(X^*, s)$ takes the range of all positive numbers and that it is monotonic, we have

$$\phi(X^*, s) \to +\infty, \text{ as } s \searrow \alpha_{X^*}. \tag{5.86}$$

https://doi.org/10.51202/9783186952035 - Generiert durch IP 216.73.216.36, am 18.01.2026, 21:11:22. © Urheberrechtlich geschützter Inhalt. Ohne gesonderte Erlaubnis ist jede urheberrechtliche Nutzung untersagt, insbesondere die Nutzung des Inhalts im Zusammenhang mit, für oder in KI-Systemen, KI-Modellen oder Generativen Sprachmodellen.

Now let us consider the following equation

$$\phi(X, s) = \frac{1}{\epsilon^*}, \tag{5.87}$$

where $\epsilon^* > 0$. Because of Eq. (5.85), we can apply the Implicit Function Theorem 2.5.1 to Eq. (5.87) in order to calculate s from X. Hence, for any $X \in \mathcal{M}_n$, the solution s of Eq. (5.87) with $s > \alpha_X$ is uniquely determined. Hence, $\forall \epsilon^* > 0$, Eq. (5.87) explicitly defines a function $s(X) : \mathcal{M}_n \to \mathbb{R}$, for $s > \alpha_X$. This function satisfies

$$\phi(X, s(X)) \equiv \frac{1}{\epsilon^*}.$$

Considering that function ϕ is sufficiently smooth, function $s(X)$ is therefore also smooth. Furthermore, from Eq. (5.86), we also know that $\forall X \in \mathcal{M}_n$,

$$s(X) \searrow \alpha_X, \text{ as } \epsilon^* \searrow 0.$$

Hence, for sufficient small ϵ^*, the implicitly defined smooth function $s(X)$ approximates the spectral abscissa of matrix X.

If we additionally consider X to be a function of variable $x \in \mathbb{R}^m$, i.e., $X = M(x)$, and if we denote the implicitly defined function $s(M(x))$ through Eq. (5.87) as $\bar{\alpha}^{\epsilon^*}_{M(x)}(x)$, then $\bar{\alpha}^{\epsilon^*}_{M(x)}(x)$ is the proposed smoothened function of $\alpha_{M(x)}(x)$ [174].

The remaining task is to evaluate the function $\bar{\alpha}^{\epsilon}_{M(x)}(x)$ and its derivatives for a given smoothing grade ϵ^* and a given point x^*. Instead of integrating Eq. (5.83) directly, Vanbiervliet et al. [174] provided a formula that can do this task more efficiently. Evaluating $\bar{\alpha}^{\epsilon^*}_{M(x)}(x)$ at $x = x^*$ means to solve the nonlinear function (5.87) for $X^* = M(x^*)$. This can be done efficiently by using standard root-finding methods, if the value of $\phi(X^*, s)$ and its gradients to X can be computed in an efficient way. It is proven by Vanbiervliet et al. [174] that

$$\begin{aligned} \phi(X, s) &= trace(P) = trace(Q), \\ \frac{\partial \phi(X, s)}{\partial s} &= -2trace(QP) = -2trace(PQ), \\ \frac{\partial \phi(X, s)}{\partial X} &= 2QP. \end{aligned} \tag{5.88}$$

P and Q are $n \times n$ symmetric real matrices, satisfying the Lyapunov equations

$$\begin{aligned} 0 &= (X - sI)P + P(X - sI)^T + I, \\ 0 &= (X^T - sI)Q + Q(X^T - sI)^T + I. \end{aligned} \tag{5.89}$$

Note that, Eq. (5.88) gives a formula to calculate the derivatives with respect to each element of X. If we consider $X = M(x)$ to be a smooth function of x, the derivatives of $\phi(M(x), s)$ with respect to x can be calculated straightforwardly by the chain rule.

The gradients of $\bar{\alpha}^{\epsilon^*}_X$ with respect to X can be evaluated from

$$\frac{\partial \bar{\alpha}^{\epsilon^*}_X}{\partial X} = \frac{QP}{trace(QP)}, \tag{5.90}$$

where Q, P satisfy the Lyapunov equations (5.89) with $s = \bar{\alpha}^{\epsilon^*}_X$. The gradients of function $\bar{\alpha}^{\epsilon^*}_{M(x)}(x)$ with respect to x can be calculated similarly by the chain rule.

https://doi.org/10.51202/9783186952035 - Generiert durch IP 216.73.216.36, am 18.01.2026, 21:11:22. © Urheberrechtlich geschützter Inhalt. Ohne gesonderte Erlaubnis ist jede urheberrechtliche Nutzung untersagt, insbesondere die Nutzung des Inhalts im Zusammenhang mit, für oder in KI-Systemen, KI-Modellen oder Generativen Sprachmodellen.

In summary, the evaluation of the smoothened spectral abscissa $\bar{\alpha}^{\epsilon^*}_{M(x)}(x)$ can be performed at the cost of solving Lyapunov equations multiple times and the evaluation of its gradients comes at the cost of solving a Lyapunov equation twice. Considering that there exist already efficient methods to solve Lyapunov equations, the proposed formulas allow an efficient evaluation of the smoothened spectral abscissa function $\bar{\alpha}^{\epsilon}_{M(x)}(x)$ and its gradients. The proposed smoothened method can be integrated into a derivative-based smooth optimization framework straightforwardly.

A method based on mollifiers

We present next another smoothing method based on mollifiers [77, 78]. This method is not restricted to only smoothing the spectral abscissa function, but that it can be applied to any locally integrable functions, such as spectral radius (2.6). However, a practical disadvantage of this method is that it requires the evaluation of a multi-dimensional integral, which is computationally demanding. To the author's knowledge, the method has been applied to approximate the feasible set of semi-infinite optimization problem and nonlinear optimization problems [77, 78], but it has not been applied to smoothen the spectral abscissa function.

Let $x \in \mathbb{R}^m$ and $||x||_2$ denote the Euclidean norm of x. The standard mollifier is a $\mathcal{C}^\infty$-function

$$\eta(x) = \begin{cases} \kappa e^{(||x||_2^2 - 1)^{-1}}, \text{if } ||x||_2 < 1, \\ 0, \text{if } ||x||_2 \geq 1, \end{cases}$$

where $\kappa > 0$ is a constant such that

$$\int\limits_{\mathbb{R}^m} \eta(x) dx = 1.$$

For $\epsilon > 0$, define

$$\eta^\epsilon(x) = \frac{1}{\epsilon^m} \eta(\frac{x}{\epsilon}),$$

and let

$$B(0, \epsilon) = \{x \in \mathbb{R}^m \mid ||x||_2 < \epsilon\}$$

be an open ball with radius ϵ. For any set, we use overline notation to denote its topological closure. It can be proven that $\eta^\epsilon(x)$ is also a $\mathcal{C}^\infty$-function. Its support $\overline{\{x \in \mathbb{R}^m | \eta^\epsilon(x) \neq 0\}}$ is a closed ball $\overline{B(0, \epsilon)}$.

For any locally integrable function $\beta : \mathbb{R}^m \to \mathbb{R}$ and $\epsilon > 0$, we define

$$\beta^\epsilon(x) = \eta^\epsilon(x) * \beta(x) = \int\limits_{\mathbb{R}^m} \eta^\epsilon(z) \beta(x - z) dz = \int\limits_{B(0,\epsilon)} \eta^\epsilon(z) \beta(x - z) dz.$$

Hence, $\beta^\epsilon(x)$ is a convolution, denoted as "$*$", of $\eta^\epsilon(x)$ and $\beta(x)$. It is proven that $\beta^\epsilon(x)$ is a $\mathcal{C}^\infty$-function (cf. Theorem 2.7 of [77]). For $\epsilon \searrow 0$, function $\beta^\epsilon(x)$ converges to function $\beta(x)$ in a certain sense.

Because the spectral abscissa function is continuous and therefore locally integrable, we can use mollifiers $\eta^\epsilon(x)$ to smoothen the spectral abscissa function. Hence,

$$\tilde{\alpha}^\epsilon(x) := \eta^\epsilon(x) * \alpha_{M(x)}(x) = \int\limits_{B(0,\epsilon)} \eta^\epsilon(z) \alpha_{M(x-z)}(x - z) dz, \tag{5.91}$$

https://doi.org/10.51202/9783186952035 - Generiert durch IP 216.73.216.36, am 18.01.2026, 21:11:22. © Urheberrechtlich geschützter Inhalt. Ohne gesonderte Erlaubnis ist jede urheberrechtliche Nutzung untersagt, insbesondere die Nutzung des Inhalts im Zusammenhang mit, für oder in KI-Systemen, KI-Modellen oder Generativen Sprachmodellen.

where $\tilde{\alpha}^{\epsilon}(x)$ denotes the spectral abscissa smoothened by mollifiers. $M : \mathbb{R}^m \to \mathcal{M}_n$, as defined before, denotes a matrix-valued function.

The first-order gradients of $\tilde{\alpha}^{\epsilon}(x)$ with respect to the j-th element x_j of x can be computed from

$$\frac{\partial \tilde{\alpha}^{\epsilon}(x)}{\partial x_j} = \int\limits_{B(0,\epsilon)} \frac{\partial \eta^{\epsilon}}{\partial z_j}(z)\alpha_{M(x-z)}(x-z)dz, \ j = 1, \cdots, m.$$

This equation is obtained by using the formula for derivatives of a general convolution. Hence, for any differentiable function $a(x)$ and any possibly non-differentiable function $b(x)$, if $c(x) = a(x) * b(x)$, we have

$$\frac{\partial c(x)}{\partial x_j} = \frac{\partial a(x)}{\partial x_j} * b(x), j = 1, \cdots, m.$$

As it can be seen from Eq. (5.91), evaluation of $\tilde{\alpha}^{\epsilon}(x)$ requires multi-dimensional integration over a closed ball $\overline{B(0,\epsilon)} \subset \mathbb{R}^m$. If the dimension m is bigger than say 10, it is computationally problematic. However, a great advantage of this method is that, it is based on a rather mild conditions, i.e., on local integrability of function $\beta(x)$, and therefore it can be applied to smoothen a broad class of non-smooth functions.

5.5 Challenges of solving the derived reactor network synthesis problem

Having reviewed all relevant optimization problems, we are ready to discuss the solution strategy to solve problem (3.44) and (4.39). For the sake of brevity, the discussion will be targeted to problem (4.39), because compared with problem (3.44) it contains additional complementarity constraints, and therefore problem (4.39) is more general and difficult to solve than problem (3.44). The proposed solution strategy can be adapted to solve problem (3.44) straightforwardly.

Problem (4.39) is a semi-infinite mixed-integer nonlinear program (MINLP) with complementarity constraints, disjunctions and robust eigenvalue constraints. Integers and disjunctions can be treated by using the methods reviewed in Section 5.1. Complementarity constraints, robust/uncertain constraints and eigenvalue constraints can be treated by using the methods reviewed in Section 5.2, 5.3 and 5.4, respectively. However, when all these features come together as in problem (4.39), the problem becomes very difficult to solve even to local optimality.

The difficulty to solve problem (4.39) is mainly due to Eq. (4.39f), which combines the features of parametric uncertainty and the non-smoothness of eigenvalue constraints. To the author's knowledge, a semi-infinite optimization (SIP) with an embedded *non-smooth* inner optimization problem has rarely been discussed in literature. Moreover, discrete decisions on integer variables, disjunctions and complementarity constraints make the solution of this optimization problem even more difficult. Problem size, i.e., the number of variables and constraints, is another practical issue when solving problem (4.39), since reactor network synthesis problems typically result in large optimization problems. Hence, it is very challenging to get even a local minimum of problem (4.39) properly.

In the next section, we propose a two-step hybrid solution approach. Solving problem (4.39) to global optimality is very challenging and out of the scope of this work.

https://doi.org/10.51202/9783186952035 - Generiert durch IP 216.73.216.36, am 18.01.2026, 21:11:22. © Urheberrechtlich geschützter Inhalt. Ohne gesonderte
Erlaubnis ist jede urheberrechtliche Nutzung untersagt, insbesondere die Nutzung des Inhalts im Zusammenhang mit, für oder in KI-Systemen, KI-Modellen oder Generativen Sprachmodellen.

5.6 A proposed two-step hybrid solution method

Because we were not able to find an existing algorithm for problem (4.39), a two-step hybrid method is pragmatically proposed here. Denote $\mathcal{P}_{\pi_\tau=\bar{\pi}_\tau}$ as the optimization problem, which is derived from problem (4.39) by fixing the uncertain parameters π_τ to their normal values $\bar{\pi}_\tau$. $\mathcal{P}_{\pi_\tau=\bar{\pi}_\tau}$ is therefore a MINLP with a deterministic eigenvalue constraint, complementarity constraints and disjunctions.

Denote $\mathcal{S}_0$ as any subset of $\{(v,w)|v=1,\cdots,n_m,w=1,\cdots,n_c\}$, i.e.,

$$\mathcal{S}_0 \subseteq \{(v,w) \mid v=1,\cdots,n_m,\ w=1,\cdots,n_c\}.$$

By "fixing the (not necessarily decentralized) control structure according to $\mathcal{S}_0$", we mean that we require the constraints

$$[K]_{v,w}=0,\ \ \forall (v,w)\in\mathcal{S}_0, \tag{5.92}$$

to hold.

Denote $\mathcal{P}_{z=z_0,\mathcal{S}_0}$ as the optimization problem, which is derived from problem (4.39) by fixing the integer variable $z=z_0$ and the control structure according to $\mathcal{S}_0$. $\mathcal{P}_{z=z_0,\mathcal{S}_0}$ is a SIP (without integer variables). We note that both $\mathcal{P}_{\pi_\tau=\bar{\pi}_\tau}$ and $\mathcal{P}_{z=z_0,\mathcal{S}_0}$ contain a non-smooth eigenvalue constraint, which results from Eq. (4.39f).

The general framework of the proposed two-step solution approach is:

- Step 1: Solve $\mathcal{P}_{\pi_\tau=\bar{\pi}_\tau}$ to obtain z_0 and $\mathcal{S}_0$ as the optimal solutions of integer variables and the control structure, respectively.

- Step 2: Solve $\mathcal{P}_{z=z_0,\mathcal{S}_0}$, initialized by the solution of step 1, to satisfy the robustness property.

We suggest to solve $\mathcal{P}_{\pi_\tau=\bar{\pi}_\tau}$ in step 1 to global optimality, or at least approximate the global minimum to obtain reasonably good solutions at the end.

We note that this two-step approach is based on the assumption that the solutions of the original problem (4.39) and problem $\mathcal{P}_{\pi_\tau=\bar{\pi}_\tau}$ are "close" in some sense: (i) they have the same solution of flowsheet and control structures, namely integer variables and set $\mathcal{S}_0$; (ii) the continuous variables of both solutions are in a certain small neighborhood. This assumption is reasonable, if the uncertainty region $[\bar{\pi}_\tau-\Delta\bar{\pi}_\tau,\bar{\pi}_\tau+\Delta\bar{\pi}_\tau]$ is not too large, because problem (4.39) reduces to problem $\mathcal{P}_{\pi_\tau=\bar{\pi}_\tau}$ when there is no parametric uncertainty. However, we must note that, if the above assumptions are not valid any more, this two-step approach may result in sub-optimal solutions.

Note also that this two-step approach closely relates to the two-phase hybrid method reviewed in Section 5.3.1. However, to keep the problem solvable in reasonable time, in the first phase only the nominal point in the uncertainty region is sampled. In the second phase, we do not use the local reduction method, but apply the normal vector approach (cf. Section 5.3.3).

The following subsections provide further detail on the solution strategies in steps 1 and 2.

https://doi.org/10.51202/9783186952035 - Generiert durch IP 216.73.216.36, am 18.01.2026, 21:11:22. © Urheberrechtlich geschützter Inhalt. Ohne gesonderte Erlaubnis ist jede urheberrechtliche Nutzung untersagt, insbesondere die Nutzung des Inhalts im Zusammenhang mit, für oder in KI-Systemen, KI-Modellen oder Generativen Sprachmodellen.

5.6.1 Step 1: Mixed-integer problem without uncertainty

There exist different strategies to solve $\mathcal{P}_{\pi_\tau=\bar{\pi}_\tau}$, depending on the selection of the combination of ways to treat integer variables, complementarity, disjunctive and eigenvalue constraints. Alternative strategies are presented in Section 5.1 to 5.4. We next talk about our choice to solve the presented case study shown in Chapter 6.

Complementarity constraints can be treated by using the penalty method, the interior point method, the SQP method or the regularization (smoothing) method (cf. Section 5.2). In this work, we apply the regularization method (cf. Eq. (5.17) in Section 5.2.2) to reformulate the complementary constraints (4.39h)-(4.39l) into a set of nonlinear constraints. More precisely, any single complementarity constraint in Eqs. (4.39h)-(4.39l),

$$0 \leq a \perp b \geq 0, a, b \in \mathbb{R},$$

is transformed into the set of nonlinear constraints

$$a \geq 0,\ b \geq 0,\ ab \leq \epsilon_c, \tag{5.93}$$

where $\epsilon_c > 0 \in \mathbb{R}$ is a small constant.

Integer variables can be treated by the Branch and Bound ($B\&B$) method (cf. Section 5.1.1) or by the complementarity-based reformulation method (cf. Section 5.2.3). The $B\&B$ method generates a search tree of subproblems through relaxing and fixing integer variables. However, if the generated subproblems are difficult or time consuming to solve, the $B\&B$ method may become costly. In this work therefore we apply the complementarity-based reformulation method, refer to Eq. (5.18), which transforms the integer variables into complementarity constraints. Denote $[z]_i \in \{0,1\}$ as the i-th element of z, $i = 1, \cdots, N + n_c$. The integer variable $[z]_i \in \{0,1\}$ is first relaxed into a continuous variable $z_i \in \mathbb{R}$, which is required to fulfill

$$0 \leq [z]_i,\ \ 1 - [z]_i \geq 0,\ \ [z]_i(1 - [z]_i) \leq \epsilon_z, \tag{5.94}$$

where $\epsilon_z > 0 \in \mathbb{R}$ is a selected small positive scalar. This reformulation method results in a single NLP without integer variables.

Disjunctions can be treated by using the big-M method or the convex hull method (cf. Section 5.1.5). In this work, we apply the big-M method, refer to Eq. (5.5), to transform the disjunctions in Eqs. (4.39m), (4.39n), (4.39p) into a set of nonlinear constraints. For any single disjunction in Eqs. (4.39m), (4.39n), (4.39p) written as

$$\begin{bmatrix} [z]_i \\ \theta_1(\cdot) \leq 0 \end{bmatrix} \vee \begin{bmatrix} \overline{[z]}_i \\ \theta_2(\cdot) \leq 0 \end{bmatrix}, [z]_i \in \{0,1\}, \tag{5.95}$$

with the general functions $\theta_1(\cdot)$ and $\theta_2(\cdot)$, the big-M method results in

$$\begin{aligned} &\theta_1(\cdot) \leq M(1 - [z]_i), \\ &\theta_2(\cdot) \leq M[z]_i, \\ &[z]_i \in \{0,1\}, \end{aligned} \tag{5.96}$$

where $M > 0 \in \mathbb{R}$ is a selected constant which is big enough.

https://doi.org/10.51202/9783186952035 - Generiert durch IP 216.73.216.36, am 18.01.2026, 21:11:22. © Urheberrechtlich geschützter Inhalt. Ohne gesonderte Erlaubnis ist jede urheberrechtliche Nutzung untersagt, insbesondere die Nutzung des Inhalts im Zusammenhang mit, für oder in KI-Systemen, KI-Modellen oder Generativen Sprachmodellen.

Last but not least, treating the non-smooth eigenvalue constraint (4.38) remains a major challenge of solving $\mathcal{P}_{\pi_\tau=\bar{\pi}_\tau}$ properly. To solve the open-loop case study (cf. Section 6.1), we assumed that the active eigenvalues of matrix $\bar{J}$ are either a single real eigenvalue or a single pair of conjugate complex eigenvalues at the local minimum. Under this assumption the eigenvalue constraint is locally smooth with respect to its arguments (cf. Section 2.2 and corollary 2.2.3) and can be treated locally as a smooth nonlinear constraint. In contrast to this direct way of treating the eigenvalue constraint to solve the closed-loop case study (cf. Section 6.2), we apply the smoothing method based on the H_2-type cost function (cf. Section 5.4.3). If we denote $\alpha^{\epsilon_e}_{\bar{J}}(\cdot)$ as the smoothened function of the original eigenvalue function $\alpha_{\bar{J}}(\cdot)$, where $\epsilon_e > 0 \in \mathbb{R}$ is a sufficiently small constant, Eq. (4.39f) evaluated at nominal points is replaced by

$$-c \geq \alpha^{\epsilon_e}_{\bar{J}}(x, e, u, \psi_c, z), \tag{5.97}$$

resulting in a reformulated MINLP in which all constraints are smooth. Compared with the direct way to treat the eigenvalue function, the smoothing method allows to employ a smooth NLP optimizer such as SNOPT [54], which can converge more robustly. However, we note that this is at the cost of higher computational effort of evaluating $\alpha^{\epsilon}_{\bar{J}}(\cdot)$ and its gradients.

To summarize, we use Eqs. (5.93), (5.94), (5.96), (5.97) to treat complementarity constraints, integer variables, disjunctions and eigenvalue constraints and therefore replace $\mathcal{P}_{\pi_\tau=\bar{\pi}_\tau}$ by $\mathcal{P}^{\epsilon_c,\epsilon_z,\epsilon_e}_{\pi_\tau=\bar{\pi}_\tau}$. For sufficiently small ϵ_c, ϵ_z and ϵ_e, we can expect that the local solutions of $\mathcal{P}^{\epsilon_c,\epsilon_z,\epsilon_e}_{\pi_\tau=\bar{\pi}_\tau}$ approximate the local solutions of $\mathcal{P}_{\pi_\tau=\bar{\pi}_\tau}$.

A good approximation of the global minimum is required in step 1, because the eigenvalue constraint (4.38) has to be satisfied not only for the nominal but also for the uncertain case. To the author's experiences, it is still very challenging to solve $\mathcal{P}_{\pi_\tau=\bar{\pi}_\tau}$ or $\mathcal{P}^{\epsilon_c,\epsilon_z,\epsilon_e}_{\pi_\tau=\bar{\pi}_\tau}$ to global optimality because of the non-smoothness of the eigenvalue constraint. To this end, we apply a multi-start strategy to solve $\mathcal{P}^{\epsilon_c,\epsilon_z,\epsilon_e}_{\pi_\tau=\bar{\pi}_\tau}$. The best obtained solution will be fed into step 2 to fix the flowsheet and the control structure, and to initialize all continuous variables.

5.6.2 Step 2: Robust optimization problem

After successfully solving $\mathcal{P}_{\pi_\tau=\bar{\pi}_\tau}$ in step 1, we denote the optimal values of the integer variables as z_0, the optimal values of the continuous variables $\mathcal{X} = (x^T, e^T, u^T, \psi_c^T, K_v^{+T}, K_v^{-T}, \hat{K}_v^T)^T$ as $\mathcal{X}_0$ and define the control structure set

$$\mathcal{S}_0 := \{(v, w) \mid [K]_{v,w}|_{\mathcal{X}=\mathcal{X}_0} = 0\}, \tag{5.98}$$

which contains the indices of the zero elements in matrix K in the optimal solution of step 1. We note that if problem $\mathcal{P}_{\pi_\tau=\bar{\pi}_\tau}$ is feasible, $\mathcal{S}_0$ must correspond to a decentralized control structure.

In step 2, we solve problem $\mathcal{P}_{z=z_0,\mathcal{S}_0}$, which is derived from the original problem (4.39) by fixing the integer variables $z = z_0$ and the control structure according to $\mathcal{S}_0$. More

https://doi.org/10.51202/9783186952035 - Generiert durch IP 216.73.216.36, am 18.01.2026, 21:11:22. © Urheberrechtlich geschützter Inhalt. Ohne gesonderte Erlaubnis ist jede urheberrechtliche Nutzung untersagt, insbesondere die Nutzung des Inhalts im Zusammenhang mit, für oder in KI-Systemen, KI-Modellen oder Generativen Sprachmodellen.

precisely, $\mathcal{P}_{z=z_0,\mathcal{S}_0}$ takes the form

$$\min_{x,e,u,\psi_c,K_v^+,K_v^-,\hat{K}_v} \quad \varphi(x,e,u,\psi_c,z_0) \tag{5.99a}$$

$$\begin{aligned} s.t.\ 0 = {} & f_i(x_i,\cdots,q_{h(i,k)}g_{h(i,k)}(x_{\bar{h}(i,k)},u_{\bar{h}(i,k)},d_{\bar{h}(i,k)}),\cdots, \\ & q_{h(i,N)}p_{sys},q_{i,1},\cdots,q_{i,N},u_i,d_i), i=1,\cdots,N, \end{aligned} \tag{5.99b}$$

$$0 = \phi_{i,r}(x_i,d_i) - \bar{y}_{i,r}, i=1,\cdots,N, r=1,\cdots,n_c^i, \tag{5.99c}$$

$$0 = e, \tag{5.99d}$$

$$0 = u - \bar{u}, \tag{5.99e}$$

$$-c \geq \alpha_{\bar{J}}(x,e,u,\psi_c,z_0), \forall \pi_\tau \in [\bar{\pi}_\tau - \Delta\bar{\pi}_\tau, \bar{\pi}_\tau + \Delta\bar{\pi}_\tau], \tag{5.99f}$$

$$\psi_c^U \geq \psi_c \geq \psi_c^L \tag{5.99g}$$

$$[K]_{v,w} = [K^+]_{v,w} - [K^-]_{v,w}, \quad v=1,\cdots,n_m, w=1,\cdots,n_c, \tag{5.99h}$$

$$[\hat{K}]_{v,w} = [K^+]_{v,w} + [K^-]_{v,w}, \quad v=1,\cdots,n_m, w=1,\cdots,n_c, \tag{5.99i}$$

$$0 \leq [K^+]_{v,w} \perp [K^-]_{v,w} \geq 0, \quad v=1,\cdots,n_m, w=1,\cdots,n_c, \tag{5.99j}$$

$$[K]_{v,w} = 0, \forall (v,w) \in \mathcal{S}_0, \tag{5.99k}$$

$$\begin{cases} \sum\limits_{j=1}^{N} \bar{q}_{i,j} + \sum\limits_{k=1}^{N} \bar{q}_{h(i,k)} > 0, \text{if } z_i|_{z=z_0} = 1,\ i=1,\cdots,N, \\ \begin{bmatrix} \bar{q}_{i,j} \\ \bar{q}_{h(i,k)} \end{bmatrix} = 0, \forall j,k=1,\cdots,N, \text{if } z_i|_{z=z_0} = 0,\ i=1,\cdots,N, \end{cases} \tag{5.99l}$$

$$\sum_{v=1}^{n_m} [\hat{K}]_{v,\varrho(i,r)} > 0, \text{if } z_{i,r}|_{z=z_0} = 1,\ \forall i=1,\cdots,N, r=1,\cdots,n_c^i. \tag{5.99m}$$

Comparing problems (4.39) and (5.99), the integer variables in the objective function (4.39a) and in the constraints (4.39b)-(4.39g) are evaluated at $z = z_0$, which results in Eqs. (5.99a)-(5.99g). Eqs. (5.99h)-(5.99j) are reproduced from Eqs. (4.39h)-(4.39j), while Eqs. (4.39k)-(4.39l) are replaced by Eq. (5.99k)[9]. Eq. (4.39m) is evaluated at $z = z_0$, which results in Eq. (5.99l). Eq. (4.39n) is evaluated at $z = z_0$, which results in Eq. (5.99m)[10]. Analogously, Eq. (4.39p) does not appear in problem (5.99), because it holds automatically[11]. Note that Eqs. (5.99l), (5.99m) need to be included to prevent any non-idle reactor or controller to become idle after solving (5.99).

Problem (5.99) is a SIP (cf. Section 5.3) with complementarity constraints (5.99h)-(5.99j) and a robust non-smooth eigenvalue constraint (5.99f). However, it does not contain any integer variables. Using the smoothing method, Eq. (5.93), we can transform the complementarity constraints into a set of nonlinear constraints. We denote the resulting SIP problem by $\mathcal{P}^{\epsilon_c}_{z=z_0,\mathcal{S}_0}$, where $\epsilon_c > 0$ is a properly selected small number.

Problem $\mathcal{P}^{\epsilon_c}_{z=z_0,\mathcal{S}_0}$ contains a robust eigenvalue constraint (5.99f), but no integer, disjunction or complementarity constraint are included. General reviews of SIP are already

---

[9] Because $\mathcal{S}_0$ corresponds to a decentralized control structure, Eq. (5.99k) ensures the satisfaction of Eqs. (4.39k), (4.39l) automatically.

[10] If $z_{i,r}|_{z=z_0} = 0$, from Eq. (5.99k), $[K]_{v,\varrho(i,r)} = 0$, $\forall v = 1,\cdots,n_m$. Because of Eqs. (5.99h)-(5.99j), $\sum_{v=1}^{n_m}[\hat{K}]_{v,\varrho(i,r)} = 0$ holds automatically.

[11] Consider reactor i^* is idle ($z_{i^*} = 0$), $i^* \in \{1,\cdots,N\}$. Because $\mathcal{S}_0$ corresponds to a feasible solution of $\mathcal{P}_{\pi_\tau=\bar{\pi}_\tau}$, $\forall v \in \Theta_{i^*}$, we have $(v,1),\cdots,(v,n_c) \in \mathcal{S}_0$. From Eqs. (5.99h)-(5.99k) we have $[\hat{K}]_{v,w} = 0$, $\forall v \in \Theta_{i^*}$, $\forall w = 1,\cdots,n_c$.

https://doi.org/10.51202/9783186952035 - Generiert durch IP 216.73.216.36, am 18.01.2026, 21:11:22. © Urheberrechtlich geschützter Inhalt. Ohne gesonderte Erlaubnis ist jede urheberrechtliche Nutzung untersagt, insbesondere die Nutzung des Inhalts im Zusammenhang mit, für oder in KI-Systemen, KI-Modellen oder Generativen Sprachmodellen.

presented in Section 5.3, including its local and global solution methods. However, due to the non-smoothness of eigenvalue constraint (5.99f), it is still very challenging to solve $\mathcal{P}^{\epsilon_c}_{z=z_0,\mathcal{S}_0}$ properly. In this work, we solve the resulting SIP $\mathcal{P}^{\epsilon_c}_{z=z_0,\mathcal{S}_0}$ to local optimality by using the NVA (cf. Section 5.3.3).

An important advantage of the proposed two-step solution approach is the initialization of the NVA by the obtained optimal continuous variables $\mathcal{X}_0$ in step 1. Typically, $\mathcal{X}_0$ locates exactly on the critical manifold, where the eigenvalue constraint (4.38) is active, i.e.,

$$\alpha_{\bar{J}}(x, e, u, \psi_c, z_0))|_{\mathcal{X}=\mathcal{X}_0} = -c.$$

To satisfy the robustness constraint (5.99f), one needs to find an optimal solution $\mathcal{X}^*$, which keeps a distance from the critical manifolds. If the uncertain region $[\bar{\pi}_\tau - \Delta\bar{\pi}_\tau, \bar{\pi}_\tau + \Delta\bar{\pi}_\tau]$ is not too large, it is reasonable to assume that $\mathcal{X}_0$ is a good approximation of $\mathcal{X}^*$.

According to our computational experience, this strategy works well for small reactor networks, say for $N \leq 3$. When large reactor networks are computed, the NVA needs more accurate initial points. Therefore, we suggest the following procedure to generate more accurate initial points. Let $\pi^k_\tau \in [\bar{\pi}_\tau - \Delta\bar{\pi}_\tau, \bar{\pi}_\tau + \Delta\bar{\pi}_\tau]$, $k = 1, \cdots, N_{sp}$, be N_{sp} random samples of the uncertain parameters. After step 1, we solve problem (5.99) with the robust constraint (5.99f) being replaced by N_{sp} nonlinear constraints

$$-c \geq \alpha^{\epsilon_e}_{\bar{J}}(x, e, u, \psi_c, z_0)), \;\; \forall \pi_\tau = \pi^k_\tau, \; k = 1, \cdots, N_{sp}. \tag{5.100}$$

The derived optimization problem is a classical NLP, but with many derived deterministic eigenvalue constraints, which can be initialized by $\mathcal{X}_0$. Denote the solution of this derived optimization problem as $\mathcal{X}^{sp}_0$. Typically, $\mathcal{X}^{sp}_0$ does not locate on the critical manifolds any more and the robust constraint (5.99f) holds approximately. Hence, $\mathcal{X}^{sp}_0$ provides a better estimate of the optimal solution of problem (5.99) than $\mathcal{X}_0$, and it is therefore used to initialize the NVA. Note that this procedure is limited by the dimension of the space of uncertain parameters.

5.7 Implementation

We implement the proposed two-step method using the software package TOMLAB [69] in the MATLAB environment on a Windows server equipped with an Intel Xeon CPU (3.47 GHz) and 96 GB RAM. TOMLAB is used as a fundamental programming environment for implementation, because it offers a flexible MATLAB-based interface to set up optimization problems, construct eigenvalue constraints and access numerical solvers. SNOPT [54] is used as a local NLP optimizer, which is called through the TOMLAB command *tomRun* to solve the derived optimization problems.

To solve the derived MINLP problem in the first step of the proposed solution method, a key task is to evaluate the eigenvalue constraint $\alpha_{\bar{J}}(x, e, u, \psi_c, z)$ in Eq. (4.39f) and its gradients, and provide them to the applied numerical solver. This task is fulfilled by using the symbolic calculation command *derivative* and the code generation command *mcode* provided by TOMLAB. The system's Jacobian matrix is computed first symbolically and then a *.m* file is generated, which returns the evaluated Jacobian matrix for given arguments x, e, u, ψ_c, z. TOMLAB command *sym2prob* is used to generate optimization problems. If the direct way of evaluating the eigenvalue constraint and its gradients is

https://doi.org/10.51202/9783186952035 - Generiert durch IP 216.73.216.36, am 18.01.2026, 21:11:22. © Urheberrechtlich geschützter Inhalt. Ohne gesonderte Erlaubnis ist jede urheberrechtliche Nutzung untersagt, insbesondere die Nutzung des Inhalts im Zusammenhang mit, für oder in KI-Systemen, KI-Modellen oder Generativen Sprachmodellen.

applied, eigenvalues and its gradients (cf. Eq. (2.12)) can be computed by using eigenvalue computation command *eig* straightforwardly. The evaluated eigenvalue function and its gradients are provided to the generated optimization problems through external functions. This way, the eigenvalue constraint can be treated by SNOPT as a typical nonlinear inequality constraint.

In contrast to the direct way of evaluating the eigenvalue constraint and its gradients, i.e., using Eq. (2.12), we also implement the smoothing method based on the H_2-type cost function (cf. Section 5.4.3). The smoothing method requires to solve a nonlinear equation (5.87) and the Lyapunov equations (5.89). The nonlinear equation (5.87) is solved by using MATLAB command *fminbnd*, while the Lyapunov equation is solved by using MATLAB command *lyap*. To our computational experiences, the applied smoothing method is quite robust, at least for $\epsilon^e \geq 10^{-15}$. The computed smoothened eigenvalue function and its gradients are provided to SNOPT by TOMLAB command *sym2prob* as external functions, just as the direct way of treating eigenvalue constraint.

The NVA approach (step 2 of the proposed solution approach) is implemented also in MATLAB, but under the assumption that the total number of critical boundaries is one (this assumption is true in our computed case studies, refer to the numerical continuation figures shown in Chapter 6.). Since the optimal solutions after implementing step 1 typically locate exactly on the critical boundaries, one can know the type and location of critical boundaries by checking the eigenvalue spectrum of the Jacobian matrix straightforwardly.

At the beginning of the study, the author implemented the NVA for the critical manifolds for both the saddle node and the Hopf bifurcations in MATLAB. However, it was realized later that the developed code is not robust for large design problems and it can not be easily automated. To simplify the implementation, we consider the following approximated robust eigenvalue constraint

$$-c \geq \alpha_{\bar{J}}^{\epsilon_e}(x(\psi_c), e(\psi_c), u(\psi_c), \psi_c, z_0) = \alpha_{\bar{J}}^{\epsilon_e}(\psi_c), \forall \pi_\tau \in [\bar{\pi}_\tau - \Delta\bar{\pi}_\tau, \bar{\pi}_\tau + \Delta\bar{\pi}_\tau], \qquad (5.101)$$

and implement the NVA for it. $\alpha_{\bar{J}}^{\epsilon_e}$ denotes the smoothened spectral abscissa of $\bar{J}$ by using the H_2-type cost function (cf. Section 5.4.3). $x(\psi_c)$, $e(\psi_c)$, $u(\psi_c)$ are functions of ψ_c, which are implicitly determined by the steady-state equations of the closed-loop model (4.7)[12]. Note that by taking ϵ_e sufficiently small, it is reasonable to assume that the critical boundaries defined by Eq. (5.101) approximate the ones of the original non-smooth robust constraint (5.99f).

A significant advantage of considering the critical boundaries defined by Eq. (5.101) is that their normal vectors can be computed straightforwardly by applying Eqs. (5.46) and (5.90), and therefore the resulting normal vector constraints (cf. Eqs. (5.47b)-(5.47d)) become simpler than the ones for the saddle node and Hopf bifurcations (cf. [119]). Moreover, the normal vectors of the robust constraint (5.101) are always pointing from the critical boundary to the feasible nominal point (cf. Lemma 5.3.4), which simplifies the programming task.

For large reactor networks, the sampling method presented in Eq. (5.100) is applied before implementing the NVA. Due to the limited computational capacity, only the vertices of the uncertainty region are sampled. In our case study, after solving the derived

---

[12] The steady states of the closed-loop model (4.7) can be denoted as $0 = \mathcal{F}(x, u, e, \psi_c)$, where $\mathcal{F} : \mathbb{R}^{n_x+n_u+n_e+n_{\psi_c}} \rightarrow \mathbb{R}^{n_x+n_u+n_e}$. Therefore, if the conditions in the IFT hold, x, u, e are implicitly-defined by ψ_c.

https://doi.org/10.51202/9783186952035 - Generiert durch IP 216.73.216.36, am 18.01.2026, 21:11:22. © Urheberrechtlich geschützter Inhalt. Ohne gesonderte Erlaubnis ist jede urheberrechtliche Nutzung untersagt, insbesondere die Nutzung des Inhalts im Zusammenhang mit, für oder in KI-Systemen, KI-Modellen oder Generativen Sprachmodellen.

optimization problem only one vertice locates on the critical boundary and therefore this vertice is used to specify the location of the critical boundary and to initialize the NVA. Initialization of the normal vector is done by using the difference between the vertice on the critical boundary and the nominal operation point in the uncertain parameter spaces.

https://doi.org/10.51202/9783186952035 - Generiert durch IP 216.73.216.36, am 18.01.2026, 21:11:22. © Urheberrechtlich geschützter Inhalt. Ohne gesonderte
Erlaubnis ist jede urheberrechtliche Nutzung untersagt, insbesondere die Nutzung des Inhalts im Zusammenhang mit, für oder in KI-Systemen, KI-Modellen oder Generativen Sprachmodellen.

6 Case study of allyl chloride production

We consider the case study of allyl chloride production, which has been already presented in the Example 3.1 of Chapter 3. Allyl chloride can be produced by means of non-catalytic chlorination of propylene in the vapor phase [129] and its reaction rates are modeled in Eq. (3.1). In this chapter we first present the computational results for open-loop (cf. Chapter 3) and then for closed-loop reactor network synthesis (cf. Chapter 4).

6.1 Open-loop reactor network design with robust stability

The open-loop reactor network synthesis problem (3.44) is solved in this section for allyl chloride production by applying the modeling procedure presented in Section 3 and the two-step solution method proposed in Section 5.6. Since we want to guarantee robust stability for the open-loop case, $c = 10^{-4}$ is chosen in Eq. (3.44c). By solving problem (3.44) we aim to find an optimal robustly stable open-loop reactor network flowsheet, including its design parameters and steady-state operating point.

6.1.1 Problem setting

The superstructure shown in Fig. 3.1 is used, which consists of both PFR and CSTR. For comparison, we consider superstructures with different numbers of N PFR and CSTR. When N is an even number, the superstructure contains an equal number of CSTR and PFR. When N is an odd number, the superstructure contains $(N-1)/2$ PFR and $(N+1)/2$ CSTR. Without loss of generality, we consider reactors i, $i = 1, \cdots, \lfloor N/2 \rfloor$, to be PFR, while reactors i, $i = \lfloor N/2 \rfloor + 1, \cdots, N$ are CSTR. The total numbers of PFR and CSTR are $\lfloor N/2 \rfloor$ and $\lceil N/2 \rceil$, respectively. All reactors are allowed to be idle or non-idle, such that one can determine the optimal number and type of used reactors in the final design. A 2-reactor network model in open-loop is already presented in Eq. (3.23). This model illustrates the modeling procedure for superstructures consisting of N reactors. Due to limited computational power, for open-loop reactor network synthesis we assume $N \leq 10$, i.e. the superstructure contains at most 5 PFR and 5 CSTR.

To optimally design an open-loop reactor network, we maximize the profit function [129]

$$\phi = C_{rev} - C_{raw} - C_{sep} - C_{equ} - C_{hc}, \tag{6.1}$$

where C_{rev}, C_{raw}, C_{sep}, C_{equ} and C_{hc} denote product revenue, raw material cost, separation cost, equipment cost and energy cost in $[\$/year]$, respectively. These terms are computed

https://doi.org/10.51202/9783186952035 - Generiert durch IP 216.73.216.36, am 18.01.2026, 21:11:22. © Urheberrechtlich geschützter Inhalt. Ohne gesonderte Erlaubnis ist jede urheberrechtliche Nutzung untersagt, insbesondere die Nutzung des Inhalts im Zusammenhang mit, für oder in KI-Systemen, KI-Modellen oder Generativen Sprachmodellen.

Table 6.1: Constants and ranges of design variables for the allyl chloride case study.

Parameter	Value	Description	Unit
p_A	0.0215	price of A	$\$/mol$
p_B	0.1287	price of B	$\$/mol$
p_C	0.0154	price of C	$\$/mol$
p_h	0.0288	price of heating	$\$/kWh$
p_c	0.0025	price of cooling	$\$/kWh$
T_{year}	3.06×10^7	annual working time	$s/year$
T_i	[450, 600]	temperature of reactor i	K
V_i	[500, 1000]	volume of CSTR i	l
L_i	[0.5, 10]	length of PFR i	m
S_i	[0, 0.1]	cross section of PFR i	m^2
T_{sys}	[300, 600]	feed temperature	K

as:

$$\begin{aligned}
C_{rev} &= p_B T_{year}[y_{sys}]_2,\\
C_{raw} &= p_A T_{year}(\sum_j q_{N+1,j}[p_{sys}]_1 - [y_{sys}]_1) + p_C T_{year}(\sum_j q_{N+1,j}[p_{sys}]_3 - [y_{sys}]_3),\\
C_{sep} &= 10^5 \cdot ([y_{sys}]_1 + [y_{sys}]_2 + [y_{sys}]_3),\\
C_{equ} &= 4.2 \times 10^6 (\sum_i V_i)^{0.63},\\
C_{hc} &= T_{year}(\sum_{i,Q_{hi}>0} p_h Q_{hi} + \sum_{i,Q_{ci}>0} p_c Q_{ci}).
\end{aligned}$$

A, B and C refer to component propylene, allyl chloride and chlorine, respectively. p_A, p_B and p_C are molar prices in $[\$/mol]$ of components A, B and C. p_h and p_c denote the prices for heating and cooling in $[\$/kWh]$, respectively. T_{year} denotes the annual operating hours in $[h]$. C_{sep} is an estimate of the annual separating cost to cover the contribution of the separation part of the process, which is not included in the reactor network. C_{sep} is proportional to the total molar flowrate of components A, B and C in the system's outlet. C_{equ} is an estimate of the annual capital cost of equipment, which is related to the total volume of the reactors. The values of all parameters are listed in Table 6.1. p_{sys} and y_{sys} refer to the system inlet and outlet of the reactor network (cf. Eqs. (3.16) and (3.15)). $q_{N+1,j}$ denotes the flowrate of the j-th outlet of the system's mixer.

The reactor network is fed with raw materials A and C. Each has a maximal flowrate of 10 $[mol/s]$, i.e.

$$\begin{aligned}
0 \leq & f_{sys}^A \leq 10\ mol/s,\\
0 \leq & f_{sys}^C \leq 10\ mol/s,
\end{aligned} \tag{6.2}$$

where f_{sys}^A and f_{sys}^C refer to the flowrates of component A and C in the system's feed in $[mol/s]$. The feed temperature T_{sys} is 300 K, and if necessary it can be heated to 600 K, i.e.

$$300 \leq T_{sys} \leq 600\ K. \tag{6.3}$$

The volume of each CSTR should be within $[500, 1000]$ l, i.e.

$$500 \leq V_i \leq 1000\ l,\ i = \lfloor N/2 \rfloor + 1, \cdots, N. \tag{6.4}$$

https://doi.org/10.51202/9783186952035 - Generiert durch IP 216.73.216.36, am 18.01.2026, 21:11:22. © Urheberrechtlich geschützter Inhalt. Ohne gesonderte Erlaubnis ist jede urheberrechtliche Nutzung untersagt, insbesondere die Nutzung des Inhalts im Zusammenhang mit, für oder in KI-Systemen, KI-Modellen oder Generativen Sprachmodellen.

Table 6.2: Uncertain parameters for the allyl chloride case study.

uncertain parameter	unit	nominal value	uncertainty
f^A_{sys}	mol/s	s.t. optimization	$\pm$ 0.2 mol/s
f^C_{sys}	mol/s	s.t. optimization	$\pm$ 0.2 mol/s
a_1	$1/s$	1.5×10^6	$\pm$ 5%
a_2	$1/s$	4.4×10^8	$\pm$ 5%
a_3	$l/mol/s$	1.0×10^2	$\pm$ 5%

Table 6.3: Open-loop process design parameters for the allyl chloride case study.

CSTR	PFR	network
V_i	L_i	f^A_{sys}, f^C_{sys}, T_{sys}
Q_{hi}	S_i	q
	Q_{hi}	z

The length and each cross section area of the PFR should be in the range of $[0.5, 10]$ m and less than 0.1 m^2, respectively, i.e.

$$\begin{aligned} 0.5 \leq L_i \leq 10\ m,\ i = 1, \cdots, \lfloor N/2 \rfloor, \\ 0 \leq S_i \leq 0.1\ m^2,\ i = 1, \cdots, \lfloor N/2 \rfloor. \end{aligned} \tag{6.5}$$

The operating temperature of both, CSTR and PFR, should be within $[450, 600]$ K, i.e.

$$450 \leq T_i \leq 600\ K,\ i = 1, \cdots, N. \tag{6.6}$$

The feed rates f^A_{sys} and f^C_{sys} of raw materials A and C and the reaction rate constants a_1, a_2 and a_3 are considered as uncertain parameters, i.e.

$$\pi_\tau = (f^A_{sys}, f^C_{sys}, a_1, a_2, a_3)^T \in \mathbb{R}^5. \tag{6.7}$$

Their nominal values and uncertainty regions are summarized in Table 6.2. f^A_{sys} and f^C_{sys} correspond to process uncertainties in the system's feeding stream. We take the assumption that these process uncertainties are subject to slow disturbances [118], i.e., the actual values of $f^A_{sys}(t)$ and $f^C_{sys}(t)$ change much slower than the time scale of the system. a_1, a_2 and a_3 correspond to model uncertainties. Their accurate values are not known exactly a priori. For an N reactor network, the total number of uncertain parameters is 5. The nominal values of f^A_{sys} and f^C_{sys} are subject to optimization, while their uncertainties are ± 0.2 mol/s. The nominal values of reaction constants a_1, a_2 and a_3 are fixed, but each of them is subject to 5% uncertainty of their nominal values.

The process design parameters of the open-loop rector network synthesis, which are the degrees of freedom ψ_o defined in Eq. (3.22) of the open-loop design problem (3.44), are summarized in Table 6.3. Process design parameters include both design parameters of individual reactors as well as design parameters related to the flowsheet structure, i.e., flowrates of interconnections, feed rates of raw materials in the system's inlet, as well as the existence of each reactor.

https://doi.org/10.51202/9783186952035 - Generiert durch IP 216.73.216.36, am 18.01.2026, 21:11:22. © Urheberrechtlich geschützter Inhalt. Ohne gesonderte Erlaubnis ist jede urheberrechtliche Nutzung untersagt, insbesondere die Nutzung des Inhalts im Zusammenhang mit, für oder in KI-Systemen, KI-Modellen oder Generativen Sprachmodellen.

6.1.2 Design results

The proposed two-step solution approach in Section 5.6 is applied to solve problem (3.44) for the allyl chloride case study. To compare the design results we also solve the nominal reactor network synthesis problem without considering the eigenvalue constraint. This problem is in the form of Eq. (3.44) but without constraint (3.44c) and parametric uncertainty. It is solved straightforwardly by using the NLP solver SNOPT. A multi-start strategy is applied to search for the global minimum.

Fig. 6.1 shows the optimal values of the objective function ϕ according to Eq. (6.1) as a function of the total number of reactors N, $N = 1, \cdots, 10$, in the superstructure. The triangular points refer to the nominal designs without eigenvalue constraints. After checking the eigenvalues of the system's Jacobian matrix, we find that all these designs are unstable. The square points refer to the optimal designs with eigenvalue constraints, which are robustly stable with respect to parameter uncertainty, i.e., the solutions of problem (3.44). The difference between the triangular and square points represents the cost of ensuring robust stability.

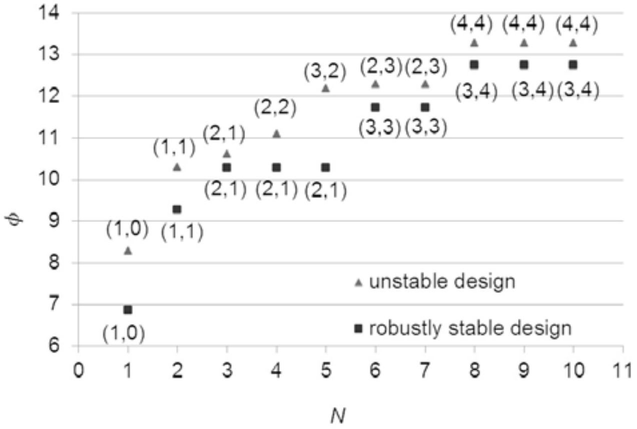


Figure 6.1: Optimal profits ϕ in $[M\$/year]$ of open-loop unstable designs and robustly stable designs for different numbers N of reactors in a superstructure. (i, j) denotes that there exist i non-idle CSTR and j non-idle PFR.

Fig. 6.1 shows that increasing the total number of reactors in the superstructure results in a higher profit. For the unstable designs (triangular points), the optimal profit increases for $N \leq 8$. For the robustly stable designs (square points), the optimal profit increases consistently for $N \leq 3$. After that, the superstructures with $N = 4, 5$ does not seem to offer any better solution than the 3-reactor network. However, if more than 6 reactors are included, a better solution can be found. This phenomenon can also be observed for designs with $N = 6, 7$ and $N = 8, 9, 10$.

https://doi.org/10.51202/9783186952035 - Generiert durch IP 216.73.216.36, am 18.01.2026, 21:11:22. © Urheberrechtlich geschützter Inhalt. Ohne gesonderte Erlaubnis ist jede urheberrechtliche Nutzung untersagt, insbesondere die Nutzung des Inhalts im Zusammenhang mit, für oder in KI-Systemen, KI-Modellen oder Generativen Sprachmodellen.

Table 6.4: Design parameters for the unstable reactor network design shown in Fig. 6.2.

var. unit	L_i $[m]$	S_i $[m^2]$	V_i $[l]$	Q_{hi} $[MJ/s]$
PFR 1	6.96	0.008	-	-0.241
PFR 2	1.58	0.014	-	-0.352
PFR 3	3.54	0.045	-	-0.142
PFR 4	1.82	0.017	-	-0.477
CSTR 5	-	-	500	0.398
CSTR 6	-	-	500	-0.121
CSTR 7	-	-	500	0
CSTR 8	-	-	500	-0.066

Table 6.5: Reactor states for the unstable reactor network design shown in Fig. 6.2. States of PFR refer to the outlet of the tube.

var. unit	propylene $[mol/l]$	allyl chloride $[mol/l]$	chlorine $[mol/l]$	temperature $[K]$
PFR 1	0.4890	1.3327	0.3544	464.3
PFR 2	0.6065	0.8679	0.5323	473.6
PFR 3	0.3640	1.9282	0.1428	450.0
PFR 4	0.4169	0.3009	0.3962	501.4
CSTR 5	0.0722	0.0172	0.0714	546.7
CSTR 6	0.2014	1.5596	0.0081	450.0
CSTR 7	0.0034	0.0179	0.0013	450.0
CSTR 8	0.1016	0.3119	0.0698	450.1

Fig. 6.1 also indicates the optimal combinations of the total numbers of used (non-idle) PFR and CSTR. (i, j) in the figure refers to the number of i non-idle CSTR and j non-idle PFR in the optimal design. For example, for the reactor network superstructure containing 4 CSTR and 3 PFR (refer to point $N = 7$ in the figure), $(2, 3)$ on the dashed line means that the final unstable design contains 2 non-idle CSTR and 3 non-idle PFR; thus, 2 CSTR are idle.

Fig. 6.2 shows the optimal unstable open-loop design for the reactor network superstructure containing $N = 10$ reactors. It is computed by solving the nominal design problem without eigenvalue constraints. The design refers to $N = 8, 9, 10$ in Fig. 6.1 of the triangular points. This is the best open-loop design results we obtained so far, which has an objective function value of 13.2665 $[M\$/year]$. However, because no eigenvalue constraint is considered, the design is unstable. As we can see from the figure, it contains 4 non-idle CSTR and 4 non-idle PFR, which are connected in a non-trivial pattern. Design parameters and states of each reactor are summarized in Tables 6.4 and 6.5, respectively.

After implementing step 1 of the proposed 2-step solution approach, the obtained optimal design is presented in Fig. 6.3. There are 4 non-idle CSTR and 3 non-idle PFR included with an objective function value of 12.7742 $M\$/year$. The design is stable, because eigenvalue constraint is considered. However, it is not robustly stable, because parametric uncertainty is not considered in this step. Design parameters and states of each reactor of this design are summarized in Tables 6.6 and 6.7, respectively.

https://doi.org/10.51202/9783186952035 - Generiert durch IP 216.73.216.36, am 18.01.2026, 21:11:22. © Urheberrechtlich geschützter Inhalt. Ohne gesonderte Erlaubnis ist jede urheberrechtliche Nutzung untersagt, insbesondere die Nutzung des Inhalts im Zusammenhang mit, für oder in KI-Systemen, KI-Modellen oder Generativen Sprachmodellen.

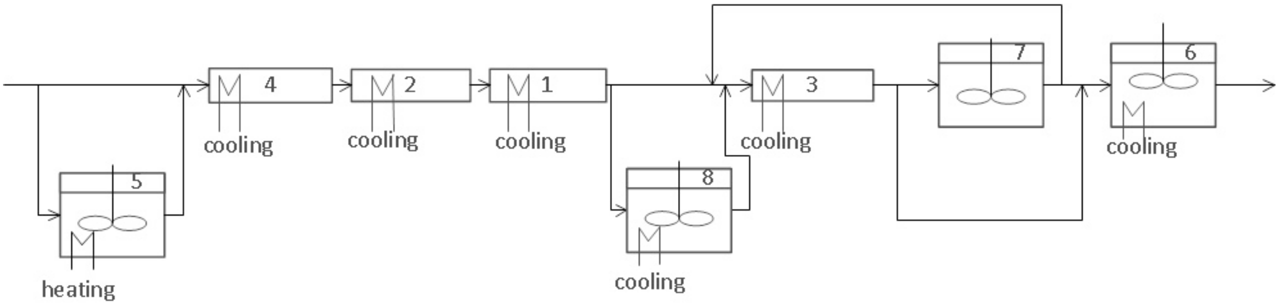


Figure 6.2: Optimal open-loop unstable design with 4 non-idle CSTR and 4 non-idle PFR. The design has an objective function value of 13.2665 $M\$/year$. 10 mol/s A and 10 mol/s C at a temperature of 300 K are fed into the reactor network. The outlet contains 0.8831 mol/s A, 6.8384 mol/s B, 0.0355 mol/s C at a temperature of 450 K. Jackets' cooling or heating is applied to all reactors except for reactor 7.

https://doi.org/10.51202/9783186952035 - Generiert durch IP 216.73.216.36, am 18.01.2026, 21:11:22. © Urheberrechtlich geschützter Inhalt. Ohne gesonderte Erlaubnis ist jede urheberrechtliche Nutzung untersagt, insbesondere die Nutzung des Inhalts im Zusammenhang mit, für oder in KI-Systemen, KI-Modellen oder Generativen Sprachmodellen.

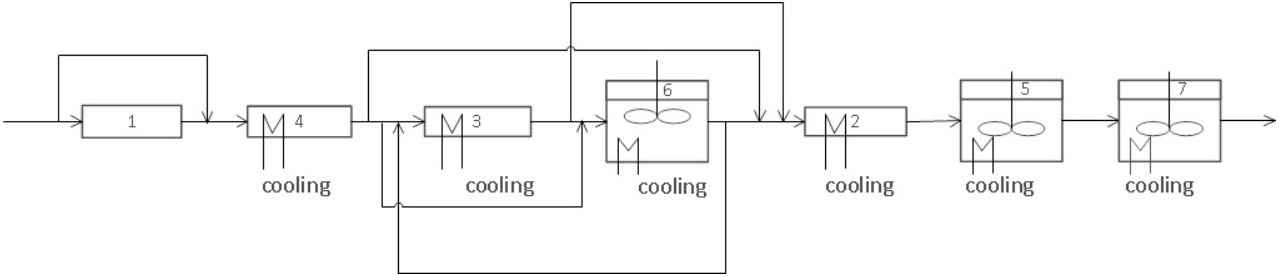


Figure 6.3: Optimal open-loop stable design, which is obtained after implementing step 1 in the proposed two-step solution approach (cf. Section 5.6). The design has 4 non-idle CSTR and 3 non-idle PFR with an objective function value of 12.7742 $M\$/year$. 10 $mol/s\ A$ and 9.9816 $mol/s\ C$ at a temperature of 504.89 K are fed into the reactor network. The outlet contains 1.0333 $mol/s\ A$, 6.6953 $mol/s\ B$, 0.1802 $mol/s\ C$ at a temperature of 450 K. Jacket cooling is applied to all reactors except for reactor 1.

https://doi.org/10.51202/9783186952035 - Generiert durch IP 216.73.216.36, am 18.01.2026, 21:11:22. © Urheberrechtlich geschützter Inhalt. Ohne gesonderte Erlaubnis ist jede urheberrechtliche Nutzung untersagt, insbesondere die Nutzung des Inhalts im Zusammenhang mit, für oder in KI-Systemen, KI-Modellen oder Generativen Sprachmodellen.

Table 6.6: Design parameters for the stable design shown in Fig. 6.3.

var. unit	L_i $[m]$	S_i $[m^2]$	V_i $[l]$	Q_{hi} $[MJ/s]$
PFR 1	0.70	0.099	-	0
PFR 2	1.92	0.068	-	-0.412
PFR 3	0.50	0.100	-	-0.130
PFR 4	1.98	0.027	-	-0.533
CSTR 5	-	-	500	-0.377
CSTR 6	-	-	500	-0.054
CSTR 7	-	-	500	-0.198

Table 6.7: Reactor states for the stable design shown in Fig. 6.3. States of PFR refer to the outlet of the tube.

var. unit	propylene $[mol/l]$	allyl chloride $[mol/l]$	chlorine $[mol/l]$	temperature $[K]$
PFR 1	0.1844	0.0249	0.1836	558.69
PFR 2	0.3036	0.6335	0.2359	478.89
PFR 3	0.0329	0.0220	0.0312	450.00
PFR 4	0.2542	0.1694	0.2409	527.65
CSTR 5	0.2936	1.1227	0.1642	450.28
CSTR 6	0.0162	0.0187	0.0153	458.64
CSTR 7	0.4034	2.6140	0.0703	450.00

To illustrate the satisfaction of the eigenvalue constraint, we visualize the critical (stability) boundaries in the uncertain parameter spaces through applying the numerical continuation toolbox Matcont [33] in Fig. 6.4. The solid lines refer to the computed critical boundaries, where the spectral abscissa is $\alpha_{\bar{J}} = -c$ with $c = 10^{-4}$. On one side of these boundaries $\alpha_{\bar{J}} < -c$, while on the other side of these boundaries $\alpha_{\bar{J}} > -c$. Since c is a very small number, crossing these boundaries will easily result in a change of stability. The solid points refer to the nominal operating point, which is obtained after implementing step 1. It locates exactly on the boundaries, which indicates that the nominal operating point is stable and the eigenvalue constraint (3.44c) is active. From the figure, we also see that the obtained nominal design is stable, but not robustly stable, because if parametric uncertainty is present the operating point may move inside the unstable region.

The robustly stable design shown in Fig. 6.5 is obtained by implementing step 2. This design consists of 3 non-idle CSTR and 4 non-idle PFR and has an objective function value of 12.4509 $[M\$/year]$. The design is robustly stable, since parametric uncertainty is considered in this step. Design parameters and states of each reactor are summarized in Tables 6.8 and 6.9, respectively. This design is as complex with non-trivial connection patterns involving CSTR and PFR as the one shown in Fig. 6.2.

Numerical continuation of the critical boundaries for the robustly stable design shown in Fig. 6.5 is implemented, refer to Fig. 6.6. Three pairs of uncertain parameters are selected for continuation, as it is done in Fig. 6.4. The solid lines refer to the computed critical boundaries, on where the spectral abscissa $\alpha_{\bar{J}} = -c$, $c = 1e-4$. On one side of these boundaries $\alpha_{\bar{J}} < -c$, while on the other side of these boundaries $\alpha_{\bar{J}} > -c$. The

https://doi.org/10.51202/9783186952035 - Generiert durch IP 216.73.216.36, am 18.01.2026, 21:11:22. © Urheberrechtlich geschützter Inhalt. Ohne gesonderte Erlaubnis ist jede urheberrechtliche Nutzung untersagt, insbesondere die Nutzung des Inhalts im Zusammenhang mit, für oder in KI-Systemen, KI-Modellen oder Generativen Sprachmodellen.

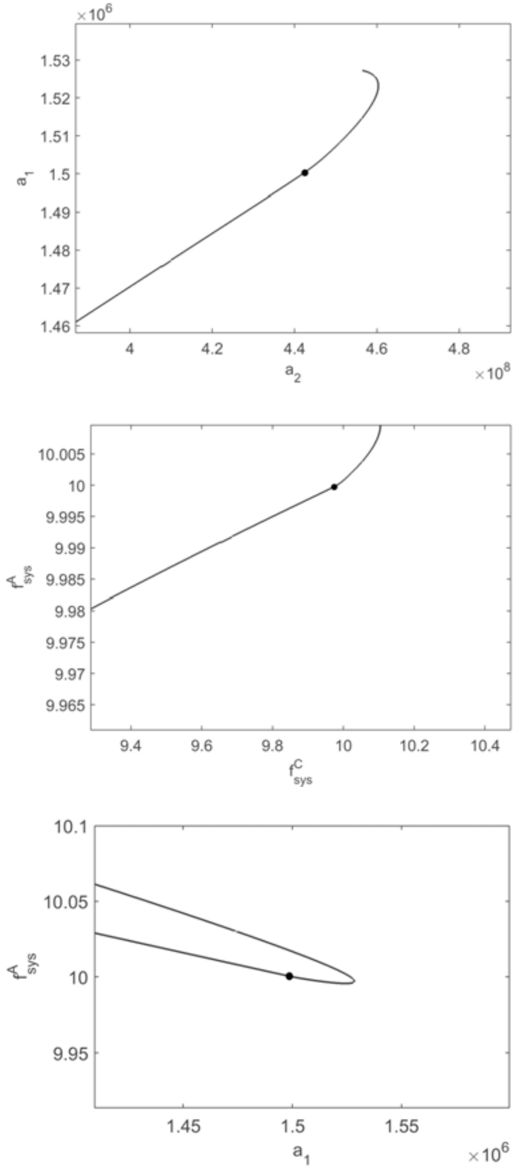


Figure 6.4: Numerical continuation study of the stability boundaries for the stable design shown in Fig. 6.3. Continuation is done for 3 selected pairs of uncertain parameters.

https://doi.org/10.51202/9783186952035 - Generiert durch IP 216.73.216.36, am 18.01.2026, 21:11:22. © Urheberrechtlich geschützter Inhalt. Ohne gesonderte
Erlaubnis ist jede urheberrechtliche Nutzung untersagt, insbesondere die Nutzung des Inhalts im Zusammenhang mit, für oder in KI-Systemen, KI-Modellen oder Generativen Sprachmodellen.

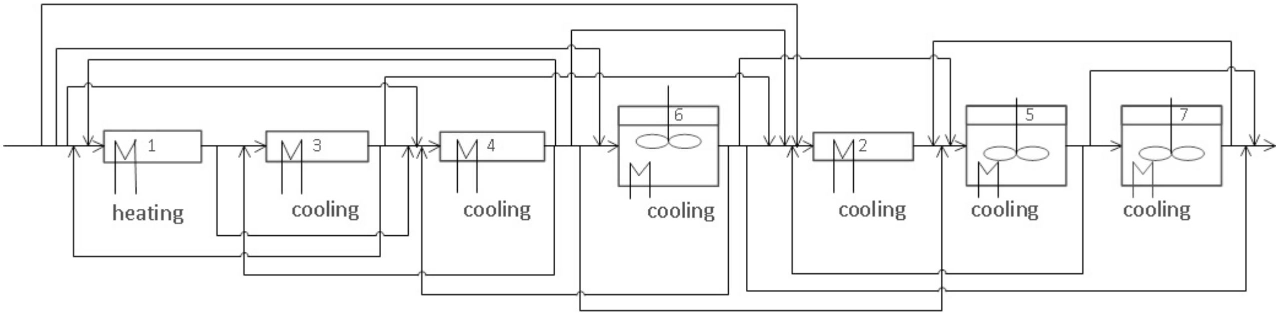


Figure 6.5: Optimal open-loop robustly stable design with 3 non-idle CSTR and 4 non-idle PFR. The design has an objective function value of 12.4509 $M\$/year$. 10 $mol/s\ A$ and 9.9582 $mol/s\ C$ at a temperature of 504.89 K are fed into the system. The outlet contains 1.1710 $mol/s\ A$, 6.5807 $mol/s\ B$, 0.2947 $mol/s\ C$ at a temperature of 450.08 K. Jacket cooling or heating is applied to all reactors.

https://doi.org/10.51202/9783186952035 - Generiert durch IP 216.73.216.36, am 18.01.2026, 21:11:22. © Urheberrechtlich geschützter Inhalt. Ohne gesonderte Erlaubnis ist jede urheberrechtliche Nutzung untersagt, insbesondere die Nutzung des Inhalts im Zusammenhang mit, für oder in KI-Systemen, KI-Modellen oder Generativen Sprachmodellen.

Table 6.8: Design parameters for the robustly stable design shown in Fig. 6.5.

var. unit	L_i $[m]$	S_i $[m^2]$	V_i $[l]$	Q_{hi} $[MJ/s]$
PFR 1	1.07	0.099	-	0.003
PFR 2	2.34	0.067	-	-0.383
PFR 3	0.50	0.100	-	-0.013
PFR 4	4.23	0.027	-	-0.516
CSTR 5	-	-	595	-0.400
CSTR 6	-	-	500	-0.148
CSTR 7	-	-	500	-0.219

solid points refer to the nominal operating point, which locates inside the stable region and keeps a distance from the boundaries. Shaded boxes refer to the uncertainty region defined in Eq. (3.44c), while the cycles refer to the overestimated uncertainty region by using the NVA. We see that, for all realization of the uncertain parameters the operating point always locates inside the stable region, which indicates that the nominal operating point is robustly stable.

We note that there are in total 5 uncertain parameters (cf. Table 6.2) and therefore the critical boundary is in 5-dimensional space. To visualize the critical boundaries in 2D, one has to select a pair of uncertain parameters each time and fix the rest 3 uncertain parameters. In Fig. 6.6, the remaining 3 uncertain parameters are fixed to their nominal values. This way, we actually project the 5-dimensional critical boundary into a selected 2-dimensional subspace. Note also that the overestimated uncertainty cycles in Fig. 6.6 do not touch the critical boundary. This is because the closest distance from the nominal operating point to the critical boundary is not in the selected 2-dimensional subspace, but along the normal vector direction $r \in \mathbb{R}^5$.

Comparing the design parameters and the reactor states between the stable and robustly stable designs (cf. Fig. 6.3 and Fig. 6.5) from step 1 and 2 of the proposed 2-step solution approach, refer to Tables 6.6, 6.7, 6.8 and 6.9, we find that both designs have very similar configurations. The design parameters are very close to each other except for the length of PFR 4 and the temperature of CSTR 5. The reactor states have also similar values except for PFR 3. This phenomenon has already been expected by the proposed 2-step solution solutions (cf. Section 5.6). That is, if the uncertainty region is not very large, the nominal operating point from step 1 will just back off of the critical boundaries, which results in an optimal solution that locates not far away from the nominal optimal solution of step 1.

Comparing the optimal unstable nominal design in Fig. 6.2 and the robustly stable design in Fig. 6.5, both designs show similar economic performance. The selectivity of propene to allyl chloride is 74.93% and 77.59% for the nominal design and the robustly stable design, respectively. More than 80% of the cost accounts for material cost C_{raw}, about 15% for separation cost C_{sep} and less than 5% for energy cost C_{heat} and equipment cost C_{equ}. Most reactors in the nominal design are cooled, which applies also to the robust design. The difference in the objective function values is 0.8156 $[M\$/year]$, which is 6.15% of the optimal profit of the nominal design. The relatively low extra cost for guaranteeing stability is accomplished by a much more complex superstructure, compared to previous work [87], and a more accurate treatment of the eigenvalue constraint.

https://doi.org/10.51202/9783186952035 - Generiert durch IP 216.73.216.36, am 18.01.2026, 21:11:22. © Urheberrechtlich geschützter Inhalt. Ohne gesonderte
Erlaubnis ist jede urheberrechtliche Nutzung untersagt, insbesondere die Nutzung des Inhalts im Zusammenhang mit, für oder in KI-Systemen, KI-Modellen oder Generativen Sprachmodellen.

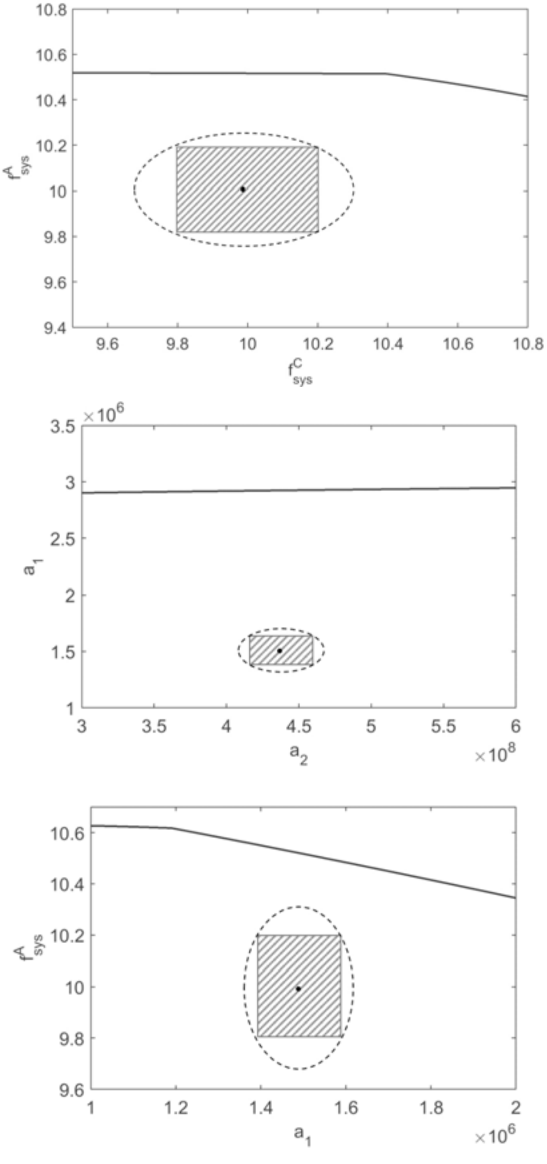


Figure 6.6: Numerical continuation study of the critical boundaries for the robustly stable design in Fig. 6.5 in selected uncertain parameter spaces.

https://doi.org/10.51202/9783186952035 - Generiert durch IP 216.73.216.36, am 18.01.2026, 21:11:22. © Urheberrechtlich geschützter Inhalt. Ohne gesonderte Erlaubnis ist jede urheberrechtliche Nutzung untersagt, insbesondere die Nutzung des Inhalts im Zusammenhang mit, für oder in KI-Systemen, KI-Modellen oder Generativen Sprachmodellen.

Table 6.9: Reactor states for the robustly stable design shown in Fig. 6.5. States of PFR refer to the outlet of the tube.

var. unit	propylene $[mol/l]$	allyl chloride $[mol/l]$	chlorine $[mol/l]$	temperature $[K]$
PFR 1	0.1496	0.0235	0.1485	555.46
PFR 2	0.2649	0.5236	0.2062	485.75
PFR 3	0.0026	0.0017	0.0025	450.00
PFR 4	0.1685	0.1116	0.1589	527.54
CSTR 5	0.2912	1.1978	0.1395	450.28
CSTR 6	0.0392	0.0268	0.0369	458.64
CSTR 7	0.4258	2.6140	0.0739	450.00

https://doi.org/10.51202/9783186952035 - Generiert durch IP 216.73.216.36, am 18.01.2026, 21:11:22. © Urheberrechtlich geschützter Inhalt. Ohne gesonderte
Erlaubnis ist jede urheberrechtliche Nutzung untersagt, insbesondere die Nutzung des Inhalts im Zusammenhang mit, für oder in KI-Systemen, KI-Modellen oder Generativen Sprachmodellen.

6.2 Simultaneous reactor network and control system design for fast response

The closed-loop reactor network synthesis problem (4.39) is solved in this section for allyl chloride production by applying the modeling procedure shown in Chapter 4 and the two-step solution method proposed in Section 5.6. Since we want to guarantee a specified response speed for the closed-loop design, $c = 0.1$ in Eq. (4.39f) is chosen. By solving problem (4.39) we aim to find an optimal closed-loop reactor network flowsheet in which the decentralized PI control structure, the process design parameters and the operating point are simultaneously determined.

6.2.1 Closed-loop reactor network modeling

The reactor network superstructure consisting of N reactors, refer to Fig. 3.1, is considered again. Each reactor is either a PFR or a CSTR. To illustrate the modeling procedure, we present a closed-loop reactor network model with only 2 CSTR. Reactor networks consisting of more than 2 reactors can be modeled in an analogous way.

The open-loop model of the 2 reactor network superstructure has already been presented in Eq. (3.23). To get its closed-loop model, the duties of the heat exchangers, namely Q_{h1} and Q_{h2}, the flowrate variables q, and the energy density $[p_{sys}]_4$ in the feed are considered as candidate MV of the network, i.e.,

$$u = (Q_{h1}, Q_{h2}, q^T, [p_{sys}]_4)^T \in \mathbb{R}^9. \tag{6.8}$$

According to Def. 4.1.1, the candidate MV of reactor 1 and 2 are Q_{h1}, $q_{1,1}$, $q_{1,2}$, $q_{2,1}$, $q_{3,1}$, and Q_{h2}, $q_{1,2}$, $q_{2,1}$, $q_{2,2}$, $q_{3,2}$, respectively. And therefore,

$$\Theta_1 = \{1, 3, 4, 5, 7\}, \text{ and } \Theta_2 = \{2, 4, 5, 6, 8\}, \tag{6.9}$$

which denote the index sets of candidate MV of reactors 1 and 2. Obviously, because $\Theta_1 \cap \Theta_2 = \{4, 5\}$, $[u]_4 = q_{1,2}$ and $[u]_5 = q_{2,1}$ are common candidate MV of both reactors.

From Eq. (4.3), we obtain

$$\pi = ([p_{sys}]_1, \cdots, [p_{sys}]_3, V_1, V_2)^T \in \mathbb{R}^5,$$

which refers to the process or equipment design parameters of the open-loop model.

We consider only the temperatures of each reactor to be measurable. Therefore $n_c^1 = 1$, $n_c^2 = 1$, i.e., each reactor has a single measurement, and the candidate CV are $y = (y_{1,1}, y_{2,1})^T = (T_1, T_2)^T \in \mathbb{R}^2$. Since $[y]_1 = y_{1,1}$ and $[y]_2 = y_{2,1}$, $\varrho(1,1) = 1$, $\varrho(2,1) = 2$ relate the subindex (i,r) of $y_{i,r}$ to the location of the w-th element $[y]_w$ in vector y (cf. Eq. (4.5)).

https://doi.org/10.51202/9783186952035 - Generiert durch IP 216.73.216.36, am 18.01.2026, 21:11:22. © Urheberrechtlich geschützter Inhalt. Ohne gesonderte Erlaubnis ist jede urheberrechtliche Nutzung untersagt, insbesondere die Nutzung des Inhalts im Zusammenhang mit, für oder in KI-Systemen, KI-Modellen oder Generativen Sprachmodellen.

Two candidate PI controllers can be now formulated according to Eqs. (4.7b), (4.7c),

$$\begin{aligned}
\dot{e}_{1,1} &= T_1 - \bar{T}_1, \\
\dot{e}_{2,1} &= T_2 - \bar{T}_2, \\
Q_{h1} &= \bar{Q}_{h1} + K_{1,1}(T_1 - \bar{T}_1 + \frac{1}{t_{1,1}} e_{1,1}) + K_{1,2}(T_2 - \bar{T}_2 + \frac{1}{t_{2,1}} e_{2,1}), \\
Q_{h2} &= \bar{Q}_{h2} + K_{2,1}(T_1 - \bar{T}_1 + \frac{1}{t_{1,1}} e_{1,1}) + K_{2,2}(T_2 - \bar{T}_2 + \frac{1}{t_{2,1}} e_{2,1}), \\
q_{1,1} &= \bar{q}_{1,1} + K_{3,1}(T_1 - \bar{T}_1 + \frac{1}{t_{1,1}} e_{1,1}) + K_{3,2}(T_2 - \bar{T}_2 + \frac{1}{t_{2,1}} e_{2,1}), \\
\vdots &= \quad \vdots \\
q_{3,2} &= \bar{q}_{3,2} + K_{8,1}(T_1 - \bar{T}_1 + \frac{1}{t_{1,1}} e_{1,1}) + K_{8,2}(T_2 - \bar{T}_2 + \frac{1}{t_{2,1}} e_{2,1}), \\
[p_{sys}]_4 &= [\bar{p}_{sys}]_4 + K_{9,1}(T_1 - \bar{T}_1 + \frac{1}{t_{1,1}} e_{1,1}) + K_{9,2}(T_2 - \bar{T}_2 + \frac{1}{t_{2,1}} e_{2,1}).
\end{aligned} \tag{6.10}$$

$e = (e_{1,1}, e_{2,1})^T$ refers to the state of the control system. $\bar{T}_1$ and $\bar{T}_2$ refer to the reference values of the candidate CV, and $\bar{Q}_{h1}, \cdots, [\bar{p}_{sys}]_4$ refer to the offset values of the candidate MV. Let

$$K = \begin{pmatrix} K_{1,1} & K_{1,2} \\ \vdots & \vdots \\ K_{9,1} & K_{9,2} \end{pmatrix} \in \mathbb{R}^{9\times 2}, \quad T = diag(1/t_{1,1}, 1/t_{2,1}) \in \mathbb{R}^{2\times 2} \tag{6.11}$$

be the proportional and the integral control gain matrices with parameters $K_{i,j} \in \mathbb{R}$, $i = 1, \cdots, 9$, $j = 1, 2$, and $t_{1,1}, t_{2,1} \in \mathbb{R}$.

Eqs. (3.23), (6.10) refer to the closed-loop model of the 2-reactor network, which specializes Eq. (4.7). The decision variables are

$$\psi_c = (\pi, \underbrace{\bar{Q}_{h1}, \bar{Q}_{h2}, \bar{q}^T, [\bar{p}_{sys}]_4}_{=\bar{u}^T}, \underbrace{\bar{T}_1, \bar{T}_2}_{=\bar{y}^T}, K_v^T, T_v^T)^T, \tag{6.12}$$

where $K_v := (K_{1,1}, \cdots, K_{9,2})^T \in \mathbb{R}^{18}$ and $T_v := (t_{1,1}, t_{2,1})^T \in \mathbb{R}^2$ concatenate all variables in K and T.

Complementarity constraints for control structure selection are derived straightforwardly by applying Eq. (4.11) to Eq. (6.11).

Idle reactors can be identified from Definition 4.1.2. According to Eqs. (3.23), (6.10), variables $\bar{q}_{1,1}$, $\bar{q}_{1,2}$, $\bar{q}_{2,1}$, $\bar{q}_{3,1}$ are used to identify the existence of reactor 1, and variables $\bar{q}_{1,2}$, $\bar{q}_{2,1}$, $\bar{q}_{2,2}$, $\bar{q}_{3,2}$ are used to identify the existence of reactor 2. Hence, Eq. (4.14) becomes

$$\begin{aligned}
&\begin{bmatrix} z_1 \\ \bar{q}_{1,1} + \bar{q}_{1,2} + \bar{q}_{2,1} + \bar{q}_{3,1} > 0 \end{bmatrix} \vee \begin{bmatrix} \bar{z}_1 \\ \bar{q}_{1,1} = \bar{q}_{1,2} = \bar{q}_{2,1} = \bar{q}_{3,1} = 0 \end{bmatrix}, \\
&\begin{bmatrix} z_2 \\ \bar{q}_{1,2} + \bar{q}_{2,1} + \bar{q}_{2,2} + \bar{q}_{3,2} > 0 \end{bmatrix} \vee \begin{bmatrix} \bar{z}_2 \\ \bar{q}_{1,2} = \bar{q}_{2,1} = \bar{q}_{2,2} = \bar{q}_{3,2} = 0 \end{bmatrix},
\end{aligned} \tag{6.13}$$

where $z_1, z_2 \in \{0, 1\}$ denote the existence of reactor 1 and 2, respectively.

https://doi.org/10.51202/9783186952035 - Generiert durch IP 216.73.216.36, am 18.01.2026, 21:11:22. © Urheberrechtlich geschützter Inhalt. Ohne gesonderte Erlaubnis ist jede urheberrechtliche Nutzung untersagt, insbesondere die Nutzung des Inhalts im Zusammenhang mit, für oder in KI-Systemen, KI-Modellen oder Generativen Sprachmodellen.

We apply Definition 4.1.3 to distinguish idle and non-idle controllers. Eq. (4.16) results in

$$\begin{aligned}
&\begin{bmatrix} z_{1,1} \\ \sum_{j=1}^{9} \hat{K}_{j,1} > 0 \end{bmatrix} \vee \begin{bmatrix} \bar{z}_{1,1} \\ \sum_{j=1}^{9} \hat{K}_{j,1} = 0 \end{bmatrix}, \\
&\begin{bmatrix} z_{2,1} \\ \sum_{j=1}^{8} \hat{K}_{j,2} > 0 \end{bmatrix} \vee \begin{bmatrix} \bar{z}_{2,1} \\ \sum_{j=1}^{8} \hat{K}_{j,2} = 0 \end{bmatrix},
\end{aligned} \tag{6.14}$$

where $z_{1,1}, z_{2,1} \in \{0, 1\}$ denote the existence of PI controller $(1, 1)$ and $(2, 1)$, respectively.

Structural constraints for a general N-reactor network are formulated by Eqs. (4.19), (4.21). Applying Eq. (4.19) to the 2-reactor network model leads to

$$\begin{aligned}
z_{1,1} &\leq z_1, \\
z_{2,1} &\leq z_2.
\end{aligned} \tag{6.15}$$

These relations guarantee that: (i) controller $(1, 1)$ must be idle, if reactor 1 is idle, and that (ii) controller $(2, 1)$ must be idle, if reactor 2 is idle.

Applying Eq. (4.21) to the 2-reactor network leads to

$$\begin{aligned}
&\begin{bmatrix} z_1 \\ \emptyset \end{bmatrix} \vee \begin{bmatrix} \bar{z}_1 \\ \hat{K}_{v,1} + \hat{K}_{v,2} = 0, \forall v \in \Theta_1 \end{bmatrix}, \\
&\begin{bmatrix} z_2 \\ \emptyset \end{bmatrix} \vee \begin{bmatrix} \bar{z}_2 \\ \hat{K}_{v,1} + \hat{K}_{v,2} = 0, \forall v \in \Theta_2 \end{bmatrix},
\end{aligned} \tag{6.16}$$

in which Θ_1 and Θ_1 are defined by Eq. (6.9). For example, if $z_1 = 0$, then all the v-th rows of matrix $\hat{K}$, $v \in \Theta_1$, are set to zero. This represents the case where the candidate MV Q_{h1}, $q_{1,1}$, $q_{1,2}$, $q_{2,1}$, $q_{3,1}$ of reactor 1 are not allowed to be manipulated.

6.2.2 Problem setting

This subsection illustrates the optimization problem (4.39) for networks with more than two reactors. The candidate CV of the N-reactor network including both CSTR and PFR are suitable temperatures in each reactor. The temperature of a CSTR is measured inside the reactor, while the temperature of a PFR is measured at its outlet. Hence, an N-reactor network has N candidate CV. The candidate MV include the flowrate variables q and the heating/cooling rates Q_h of each reactor and the feed temperature T_{sys}. Because an N-reactor network has $N(N + 1)$ flowrate variables and N heat exchangers, we have $N(N + 1) + N + 1$ candidate MV in total. For a 6-reactor network, the total number of candidate CV is 6 and the total number of candidate MV is 49, which already leads to a large number of control structure alternatives.

The reactor network is fed with raw materials A and C. Each has a maximal flowrate of 10 mol/s, refer to Eq. (6.2). The feed temperature T_{sys} $[K]$ is 300 K and, if necessary, it can be heated to 600 K, refer to Eq. (6.3). The volume V_i of the CSTR i, the length L_i of the PFR i and their cross section areas S_i are bounded as in the open-loop case, refer to Eqs. (6.4), (6.5). The operating temperature T_i of both, CSTR and PFR, should be within $[450, 600]$ K, refer to Eq. (6.6). We maximize the same profit function (6.1) which

https://doi.org/10.51202/9783186952035 - Generiert durch IP 216.73.216.36, am 18.01.2026, 21:11:22. © Urheberrechtlich geschützter Inhalt. Ohne gesonderte Erlaubnis ist jede urheberrechtliche Nutzung untersagt, insbesondere die Nutzung des Inhalts im Zusammenhang mit, für oder in KI-Systemen, KI-Modellen oder Generativen Sprachmodellen.

Table 6.10: Process and control design parameters of the simultaneous design.

CSTR i	PFR i	controller	network
V_i	L_i	K	f_{sys}^A
$\bar{Q}_{hi}$	S_i	T	f_{sys}^C
	$\bar{Q}_{hi}$	K^+	$\bar{T}_{sys}$
		K^-	$\bar{q}$
		$\hat{K}$	z

has been used in the open-loop case. The objective function comprises product revenue, material cost, separation cost, equipment cost and energy cost. The feed rates f_{sys}^A and f_{sys}^C of raw materials A and C and the reaction rate constants a_1, a_2 and a_3 are considered as uncertain parameters, as for the open-loop case, refer to Eq. (6.7) and Table 6.2.

The process and control design parameters of the closed-loop reactor network are summarized in Table 6.10. Design parameters of CSTR i, $i = \lfloor N/2 \rfloor + 1, \cdots, N$, include the reactor volume V_i and the heat exchange duty $\bar{Q}_{hi}$. Design parameters of PFR i, $i = 1, \cdots, \lfloor N/2 \rfloor$, include reactor length L_i, cross section area S_i and the offset value of the heat exchange duty $\bar{Q}_{hi}$. The control design parameters include the elements of the control gain matrices K and T and of the auxiliary matrices K^+, K^- and $\hat{K}$ for the determination of the decentralized control structure. There are other design parameters, including the flowrate of interconnections $\bar{q}$, the flowrate of the raw materials f_{sys}^A, f_{sys}^C, the temperature of the feed $\bar{T}_{sys}$ as well as the integers related to the existence of each reactor and controller.

For comparison, we apply the proposed method to design reactor networks with different numbers N of reactors included in the reactor network superstructure. To limit the computational effort, we set $N \leq 6$, i.e., the largest reactor network superstructure contains at most 3 PFR and 3 CSTR.

6.2.3 Results and discussion

Fig. 6.7 shows the optimal objective values ϕ (cf. Eq. (6.1)) for a varying number of reactors. The triangular symbols refer to open-loop (unstable) design results, which are obtained by maximizing ϕ subject to the open-loop reactor network model (4.1) (cf. Fig. 6.1). The square symbols refer to the simultaneous closed-loop design results with guaranteed robustness. $c = 0.1$ is chosen in Eq. (4.39f), which not only results in robust stability but also in a specified response speed. According to Table 2.1, the estimated decay time is $t_d = 46$ $[s]$. Triplets (i, j, k) in the figure indicate how many non-idle CSTR, non-idle PFR and non-idle PI controllers are included in the final design. As we can see from the figure that an increase of the total number of reactors leads to higher profits for both, the open-loop and the closed-loop designs. However, for a given N there is always a gap between the triangular and square symbols, which indicates the profit loss for ensuring stability and a specified response speed.

Fig. 6.8 presents the closed-loop design for a 6-reactor network superstructure (3 PFR and 3 CSTR). The design contains 3 non-idle PFR, 1 non-idle CSTR and 4 non-idle PI controllers. Flowrates and temperatures of the feed and product streams are listed in Table

https://doi.org/10.51202/9783186952035 - Generiert durch IP 216.73.216.36, am 18.01.2026, 21:11:22. © Urheberrechtlich geschützter Inhalt. Ohne gesonderte Erlaubnis ist jede urheberrechtliche Nutzung untersagt, insbesondere die Nutzung des Inhalts im Zusammenhang mit, für oder in KI-Systemen, KI-Modellen oder Generativen Sprachmodellen.

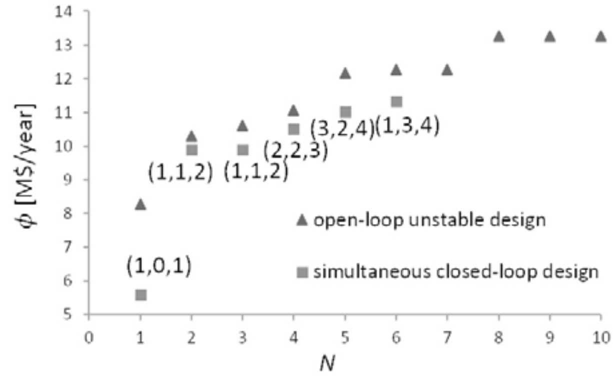


Figure 6.7: Optimal design of reactor network superstructures in open- and closed-loop. Triangular symbols refer to open-loop unstable designs. Square symbols refer to simultaneous closed-loop designs with robustly guaranteed response speed ($c = 0.1$ in Eq. (4.39f)). Triplets (i, j, k) indicate i non-idle CSTR, j non-idle PFR and k non-idle PI controllers in the closed-loop designs.

Table 6.11: Flowrates and temperatures of the feed and product streams for the closed-loop design of a 6-reactor network superstructure (3 PFR and 3 CSTR).

variable	value	unit
f_{sys}^{A}	10	mol/s
f_{sys}^{C}	10	mol/s
T_{sys}	538.51	K
f_{out}^{A}	2.79	mol/s
f_{out}^{B}	5.67	mol/s
f_{out}^{C}	2.03	mol/s
T_{out}	450	mol/s

6.11. State variables and design parameters of all reactors are listed in Table 6.12. The controller gains of all PI controllers are listed in Table 6.13.

To show that the closed-loop design in Fig. 6.8 satisfies the robust eigenvalue constraint (4.39f), we visualize the location of the critical boundaries in the space of uncertain parameters f_{sys}^{A} and f_{sys}^{C} in Fig. 6.9, and of a_1 and a_2 in Fig. 6.10. Note that there are 5 uncertain parameters in total (cf. Eq. (6.7)). Fig. 6.9 and 6.10 are projections of the critical boundary in 5-dimensional space into the selected 2-dimensional subspace. Certain parameters and the other 3 uncertain parameters are fixed to their nominal values.

In Fig. 6.9 the solid black point inside the shaded rectangular represents the nominal operating point. The shaded rectangular around it refers to the uncertainty region defined by Eq. (4.39f). The dashed ellipse refers to an over-estimated uncertainty region used in the NVA. The solid curve represents the critical boundary, at which the spectral abscissa of matrix $\bar{J}$ equals exactly $-c$. One can check that, points above this critical boundary

https://doi.org/10.51202/9783186952035 - Generiert durch IP 216.73.216.36, am 18.01.2026, 21:11:22. © Urheberrechtlich geschützter Inhalt. Ohne gesonderte Erlaubnis ist jede urheberrechtliche Nutzung untersagt, insbesondere die Nutzung des Inhalts im Zusammenhang mit, für oder in KI-Systemen, KI-Modellen oder Generativen Sprachmodellen.

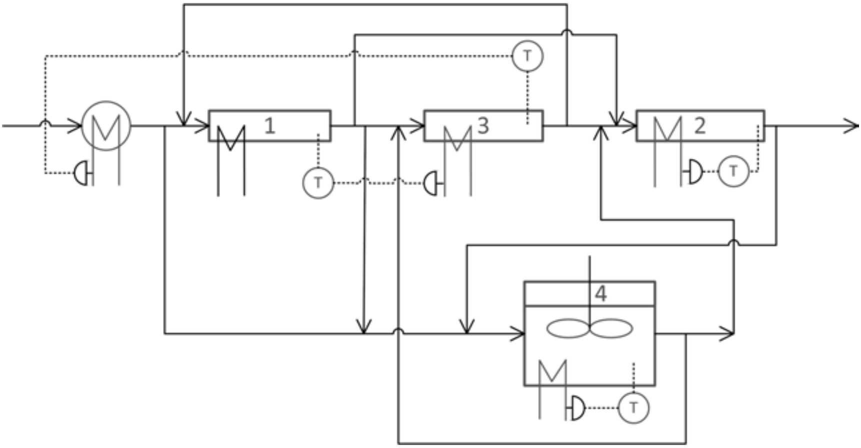


Figure 6.8: The optimal closed-loop design from the 6-reactor network superstructure with guaranteed robust response speed. This design achieves an objective function value of 11.04 $M\$/year$ and it has 3 non-idle PFR, 1 non-idle CSTR and 4 non-idle PI controllers. 10 mol/s A and 10 mol/s C with temperature of 538.51 K are fed into the system. The product stream contains 2.79 mol/s A, 5.67 mol/s B, 2.03 mol/s C at temperature 450 K.

Table 6.12: State variables and design parameters of all reactors shown in Fig. 6.8.

var. unit	c_A mol/l	c_B mol/l	c_C mol/l	T K	L_i m	S_i m^2	V_i l	Q_{hi} MJ/s
PFR 1	0.1885	0.1169	0.1793	525.43	1.77	0.0612	-	-0.659
PFR 2	0.5307	1.0764	0.3857	450	0.79	0.0600	-	-0.639
PFR 3	0.0372	0.0418	0.0324	453.99	0.50	0.0050	-	-0.100
CSTR 4	0.0157	0.0251	0.0125	450	-	-	500	-0.023

satisfy $\alpha_{\bar{J}} < -c$, while points below this critical boundary satisfy $\alpha_{\bar{J}} > -c$. Therefore, to guarantee the satisfaction of the robust eigenvalue constraint (4.39f) one needs to make sure that the solid black point locates above the critical boundary and keeps a distance from it. As we can see from the figure, for the presented nominal operating point the eigenvalue constraint $\alpha_{\bar{J}} < -c$ holds for all realizations of uncertain parameters f_{sys}^{A} and f_{sys}^{C} in the uncertainty region. Similar discussion applies to Fig. 6.10.

To verify the design results, simulation studies of the closed-loop design shown in Fig. 6.8 have been carried out by using the closed-loop model (4.7). To demonstrate the guaranteed robust dynamic properties, we select a number of operating points near the nominal operating point by randomly choosing $N_{rd} = 10$ different values of uncertain parameters π_τ in the uncertainty region. According to Eq. (4.39f), cf. also [119], all these operating points should satisfy Eq. (4.38), i.e., all these points have the specified dynamic properties. Disturbances are applied in simulations for all selected operating points to check whether

https://doi.org/10.51202/9783186952035 - Generiert durch IP 216.73.216.36, am 18.01.2026, 21:11:22. © Urheberrechtlich geschützter Inhalt. Ohne gesonderte
Erlaubnis ist jede urheberrechtliche Nutzung untersagt, insbesondere die Nutzung des Inhalts im Zusammenhang mit, für oder in KI-Systemen, KI-Modellen oder Generativen Sprachmodellen.

Table 6.13: Parameters for PI controllers shown in Fig. 6.8. TC_i, $i = 1, \cdots, 4$, refer to the temperature controllers of reactor i. CV_i and MV_i refer to the control and manipulated variables of controller TC_i. k_i and t_i refer to the proportional and integral gains of controller TC_i.

TC_i	CV_i	MV_i	k_i	t_i
TC_1	T_1 $[K]$	Q_{h3} $[J/s]$	3.9e2	6.169
TC_2	T_2 $[K]$	Q_{h2} $[J/s]$	4.9e4	2.239
TC_3	T_3 $[K]$	T_{sys} $[K]$	1.2e-2	15.74
TC_4	T_4 $[K]$	Q_{h4} $[J/s]$	1.8e2	0.679

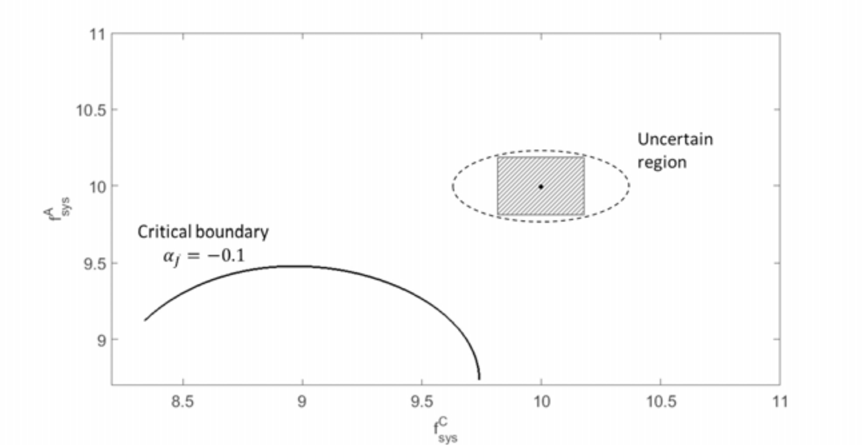


Figure 6.9: Nominal operating point, uncertainty region and critical boundary of the optimal closed-loop design are visualized in the space of uncertain parameters f^A_{sys} and f^C_{sys}.

the specified dynamic properties are fulfilled. We assume that the feed flowrates f^A_{sys} and f^C_{sys}, the feed temperature T_{sys} and the heat exchange rates $Q_{h1}, \cdots, Q_{h4}$ are subject to rectangular disturbances.

$$\begin{aligned} df^A_{sys}(t) &= \begin{cases} 5\% f^{A,0}_{sys}, \text{ if } 10 \le t \le 11 \text{ [s]} \\ 0, \text{ otherwise} \end{cases}, \\ df^C_{sys}(t) &= \begin{cases} 5\% f^{C,0}_{sys}, \text{ if } 10 \le t \le 11 \text{ [s]} \\ 0, \text{ otherwise} \end{cases}, \\ dT_{sys}(t) &= \begin{cases} 2\ [K], \text{if } 10 \le t \le 11 \text{ [s]} \\ 0, \text{ otherwise} \end{cases}, \\ dQ_{hi}(t) &= \begin{cases} 2\% Q^0_{hi}, \text{ if } 10 \le t \le 11 \text{ [s]} \\ 0, \text{ otherwise} \end{cases}, \quad i = 1 \cdots, 4. \end{aligned} \tag{6.17}$$

https://doi.org/10.51202/9783186952035 - Generiert durch IP 216.73.216.36, am 18.01.2026, 21:11:22. © Urheberrechtlich geschützter Inhalt. Ohne gesonderte Erlaubnis ist jede urheberrechtliche Nutzung untersagt, insbesondere die Nutzung des Inhalts im Zusammenhang mit, für oder in KI-Systemen, KI-Modellen oder Generativen Sprachmodellen.

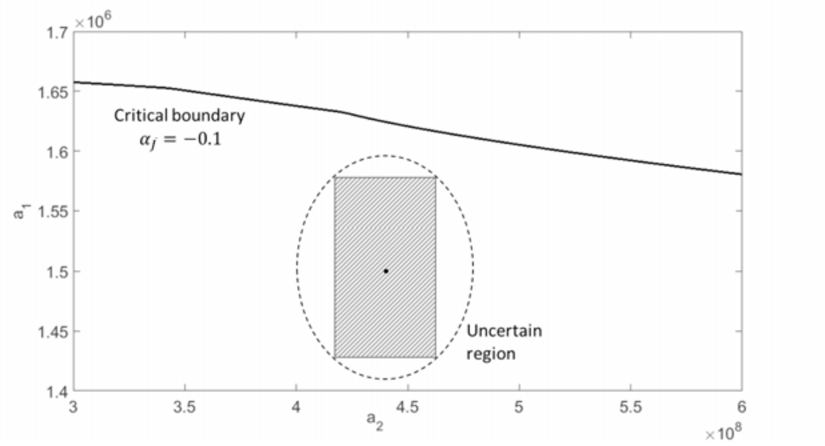


Figure 6.10: Nominal operating point, uncertainty region and critical boundary of the optimal closed-loop design are visualized in the space of uncertain parameters a_1 and a_2.

Simulation results are presented in Fig. 6.11. The system responds in a similar way for the selected operating points, as it is expected. We note that the product outlet temperature T_{out} does not change much for different values of the uncertain parameters, because T_{out} is controlled by a PI controller. Note also that the closed-loop system needs about 60 seconds to settle down, which nicely corresponds to the estimated decay time corresponding to $c = 0.1$ (cf. Table 2.1).

6.2.4 Comparison with established sequential design

To demonstrate the power of the suggested simultaneous design, at least for the presented case study, we apply the established sequential control system design to an open-loop unstable 6-reactor network. This open-loop unstable design is shown in Fig. 6.12 (removing all control loops) and it is derived by maximizing the same objective function (4.39a) but subject only to steady states of the open-loop system (4.1) and feasibility constraint (4.39g). No eigenvalue constraints or parametric uncertainties are considered. The open-loop unstable design corresponds to the triangular symbol at $N = 6$ in Fig. 6.7 with an objective function value of 12.03 $M\$/year$. 10 $mol/s\ A$ and 10 $mol/s\ C$ with temperature of 490.7 K are fed into the system. The outlet contains 1.26 $mol/s\ A$, 6.50 $mol/s\ B$, 0.52 $mol/s\ C$ at temperature of 450 K.

We linearize the open-loop model at the optimal unstable point to obtain the linearized system

$$\begin{aligned}\Delta\dot{x} &= A\Delta x + B\Delta u,\\ \Delta y &= C\Delta x.\end{aligned} \tag{6.18}$$

https://doi.org/10.51202/9783186952035 - Generiert durch IP 216.73.216.36, am 18.01.2026, 21:11:22. © Urheberrechtlich geschützter Inhalt. Ohne gesonderte
Erlaubnis ist jede urheberrechtliche Nutzung untersagt, insbesondere die Nutzung des Inhalts im Zusammenhang mit, für oder in KI-Systemen, KI-Modellen oder Generativen Sprachmodellen.

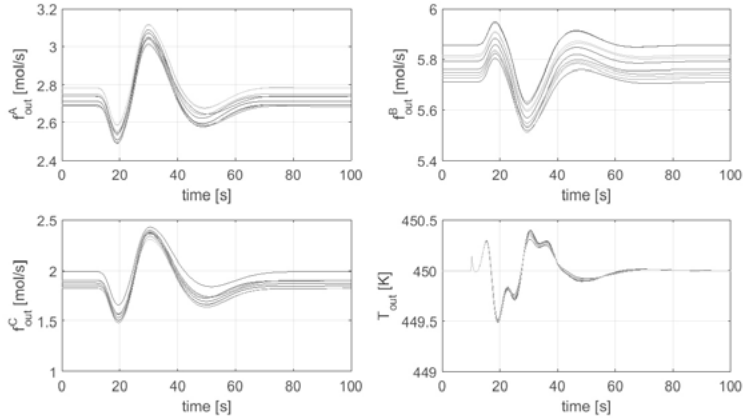


Figure 6.11: Simulation study of the simultaneous closed-loop design shown in Fig. 6.8. Disturbances in Eq. (6.17) are applied to the designed system for $N_{rd} = 10$ randomly selected values of uncertain parameters. f^A_{out}, f^B_{out}, and f^C_{out} refer to the flowrates of components A, B, C in the product outlet. T_{out} refers to the temperature of the product.

$\Delta x(t)$, $\Delta u(t)$ and $\Delta y(t)$ refer to the differences of $x(t)$, $u(t)$ and $y(t)$ from the operating point. We couple the open-loop model (6.18) with n_c candidate PI controllers,

$$\dot{e} = \Delta y, \quad \Delta u = K(\Delta y + Te), \tag{6.19a}$$

where $e \in \mathbb{R}^{n_c}$, $K \in \mathbb{R}^{n_m \times n_c}$ and $T \in \mathbb{R}^{n_c \times n_c}$ refer to the states, proportional and integral gain matrices of the controllers. The Jacobian matrix of the obtained closed-loop system is

$$J = \begin{pmatrix} A + BKC & BKT \\ C & 0 \end{pmatrix}. \tag{6.20}$$

We formulate and solve the optimization problem

$$\begin{aligned}
\min_{K_v, K_v^+, K_v^-, \hat{K}_v, T_v, z_c} \quad & \alpha_{\bar{J}}(K_v, T_v, z_c) && (6.21a)\\
s.t. \quad & Eqs.\ (4.11), (4.16), && (6.21b)\\
& K_v^{min} \leq K_v \leq K_v^{max}, && (6.21c)\\
& T_v^{min} \leq T_v \leq T_v^{max}, && (6.21d)\\
& z_c \in \{0, 1\}^{n_c}. && (6.21e)
\end{aligned}$$

where

$$\bar{J} := J - M \begin{pmatrix} 0 & 0 \\ 0 & I_0 - diag(z_c) \end{pmatrix}$$

https://doi.org/10.51202/9783186952035 - Generiert durch IP 216.73.216.36, am 18.01.2026, 21:11:22. © Urheberrechtlich geschützter Inhalt. Ohne gesonderte Erlaubnis ist jede urheberrechtliche Nutzung untersagt, insbesondere die Nutzung des Inhalts im Zusammenhang mit, für oder in KI-Systemen, KI-Modellen oder Generativen Sprachmodellen.

refers to the Jacobian considering only non-idle PI controllers (cf. Eq. (4.36) and Lemma 4.2.1). $z_c \in \{0,1\}^{n_c}$ is a vector of integer variables denoting the existence of PI controllers as defined before. $I_0 \in \mathbb{R}^{n_c \times n_c}$ is an identity matrix. K_v, K_v^+, K_v^-, $\hat{K}_v$ and T_v denote the variables in matrices K, K^+, K^-, $\hat{K}$ and T, respectively, as introduced before. K_v^{min}, K_v^{max}, T_v^{min} and T_v^{max} refer to the lower and upper bounds of K_v and T_v. Eq. (4.11) defines the decentralized control structure and Eq. (4.16) specifies idle and non-idle controllers.

Problem (6.21) is a mixed-integer problem with complementarity constraints, disjunctions and a non-smooth eigenvalue objective function. It is simpler than problem (4.39), since nonlinear steady-state constraints (4.39b) and parametric uncertainty π_τ do not appear. We apply the solution techniques presented in Section 5.6.1. In particular, Eqs. (5.93), (5.94), (5.5), (5.97) are used to treat complementarity constraints, integer variables, disjunctions and eigenvalue constraints to transform problem (6.21) into an approximate smooth NLP.

If problem (6.21) can be solved to global optimality, one can straightforwardly check whether there exists a decentralized control structure satisfying Eq. (4.38) for the given open-loop unstable operating point. Here we again apply a multi-start strategy to approximate the global minimum. Fig. 6.12 shows the best obtained solution of (6.21) for a random choice of 10^4 starting points. The spectral abscissa of the optimal design shown in Fig. 6.12 is -0.0089. For the open-loop operating point there does not seem to exist a decentralized PI control system such that the performance specified by Eq. (4.38) for $-c = -0.1$ can be reached. In other words, if the established sequential design approach is applied, one may not be able to find any decentralized control structure satisfying the specified response speed.

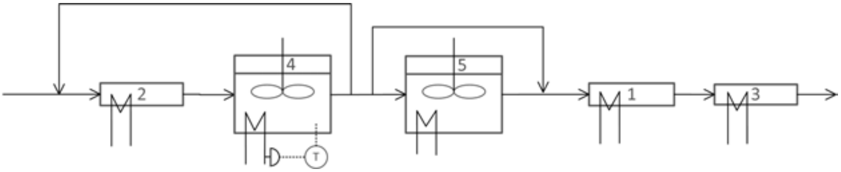


Figure 6.12: Closed-loop design obtained by a sequential design procedure. The operating point is fixed to the open-loop unstable design, which corresponds to the triangular point $N = 6$ in Fig. 6.7.

To compare the system's response of the simultaneous design shown in Fig. 6.8 and the sequential design in Fig. 6.12, another simulation study has been carried out. The disturbances defined in Eq. (6.17) are applied to the design obtained by the sequential approach. We see that the system needs about 800 s to settle down, which approximates the estimated decay time in Table 2.1 with the same order of magnitude. Compared with simulation results presented in Fig. 6.11, the response of the closed-loop system resulting from sequential design is apparently much slower.

https://doi.org/10.51202/9783186952035 - Generiert durch IP 216.73.216.36, am 18.01.2026, 21:11:22. © Urheberrechtlich geschützter Inhalt. Ohne gesonderte Erlaubnis ist jede urheberrechtliche Nutzung untersagt, insbesondere die Nutzung des Inhalts im Zusammenhang mit, für oder in KI-Systemen, KI-Modellen oder Generativen Sprachmodellen.

Table 6.14: State variables and design parameters of each reactors in Fig. 6.12. The state variables of PFR refer to the outlet of the tube.

var.	c_A	c_B	c_C	T	L_i	S_i	V_i	Q_{hi}
unit	mol/l	mol/l	mol/l	K	m	m^2	l	MJ/s
PFR 1	0.2599	0.6118	0.2045	450	2.27	0.10	-	-0.386
PFR 2	0.1261	0.0903	0.1209	450	10	0.10	-	-0.874
PFR 3	0.1424	0.7346	0.0596	450	10	0.10	-	-0.192
CSTR 4	0.0368	0.0325	0.0347	529.93	-	-	1000	-0.090
CSTR 5	0.0516	0.0480	0.0484	450	-	-	1000	-0.196

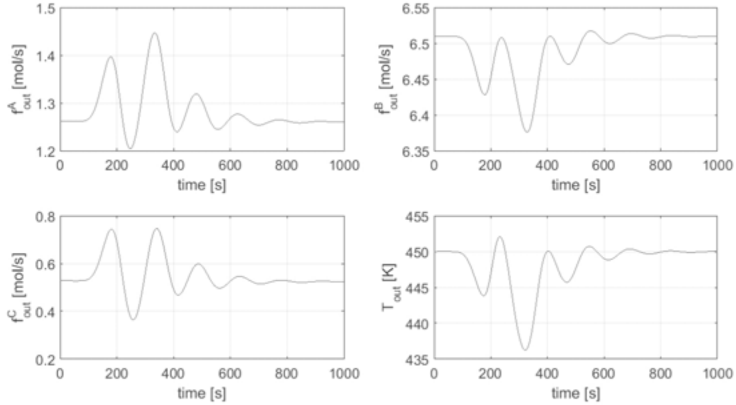


Figure 6.13: Simulation study of the closed-loop design shown in Fig. 6.12 by applying the disturbances defined in Eq. (6.17). f^A_{out}, f^B_{out}, and f^C_{out} refer to the flowrates of component A, B, C in the system's outlet. T_{out} refers to the temperature of the system's outlet.

6.3 Computational experience

Although we have proposed the 2-step solution strategy in Section 5.6, which decomposes the task of solving problem (3.44) or (4.39) into a two-step sequential procedure, solving each derived subproblem for large reactor networks is still not a trivial task. This section summarizes the computational experience gained during solving problems (3.44) and (4.39).

In the first step of the proposed solution approach, a MINLP problem is solved without considering parametric uncertainty. For both the open-loop and the closed-loop synthesis problems, $M = 1000$ is chosen for the reformulated eigenvalue constraint (3.44c), (4.39f) and integer variables are reformulated by using Eq. (5.94), in which $\epsilon^z = 10^{-6}$ is selected. Disjunctions are reformulated through the big-M method (5.96), in which $M = 1000$ is chosen.

https://doi.org/10.51202/9783186952035 - Generiert durch IP 216.73.216.36, am 18.01.2026, 21:11:22. © Urheberrechtlich geschützter Inhalt. Ohne gesonderte Erlaubnis ist jede urheberrechtliche Nutzung untersagt, insbesondere die Nutzung des Inhalts im Zusammenhang mit, für oder in KI-Systemen, KI-Modellen oder Generativen Sprachmodellen.

For the open-loop reactor network synthesis, the eigenvalue constraint (3.44c) is assumed to be locally smooth with respect to its arguments (cf. Section 2.2 and Corollary 2.2.3). This allows the eigenvalue constraint to be treated as a smooth constraint locally. For the closed-loop reactor network synthesis, we apply the smoothing method based on the H_2-type cost function (cf. Section 5.4.3 and Eq. (5.97)). A multiple starting strategy is applied for each computed superstructure with at least 50 samples.

To our computational experiences, the convergence of the applied NLP solver in step 1 is more robust, if the problem is initialized from a steady state of the reactor network model (3.44b) or (4.39b). This is probably because of the discretization of PFR, which generates a large number of nonlinear constraints (we have found that the convergence behavior of the NLP solver is more robust for reactor network synthesis problems with only CSTR). To initialize the problem, different (local) optimal solutions of the open-loop unstable designs are used to initialize the states and design parameters of the reactors, the flowrates of connections and the system parameter p_{sys}. Other design parameters, e.g. the control gain and the integer variables, are selected to be initialized randomly. We note that the eigenvalue constraint may get violated if such initialization strategy is applied. However, according to our computational experiences, SNOPT converges very often. For a reactor network superstructure with $N = 10$ reactors, solving the derived optimization problem of step 1 for the open-loop case takes typically less than 2 hours. For a reactor network superstructure with $N = 6$ reactors, solving the derived optimization problems of step 1 for the closed-loop case takes typically less than 5 hours.

In the second step, the NVA is applied to solve problem (3.44) or (4.39) with fixed reactor network and control structure. The optimal solution of the first step is used to initialize the NVA (cf. Section 5.6.2). The location and type of the critical boundaries can be checked straightforwardly, because the optimal solution from the first step typically locates on the critical boundaries (cf. e.g. Fig. 6.4).

The current implementation of the normal vector approach is based on Eq. (5.101). Compared to the first step, the second step is computationally much more demanding, which may take up to 2 days to converge for large reactor networks. A first reason is that functions $x(\psi_c)$, $e(\psi_c)$ and $u(\psi_c)$ in Eq. (5.101) are implicitly defined. Therefore, the evaluation of these functions and their gradients need to solve the nonlinear steady-state equations of the closed-loop model (4.7). Furthermore, the computation of the normal vectors of constraint (5.101) needs the evaluation of the partial gradients $\partial\alpha_{\bar{J}}^{\epsilon_e}(\psi_c)/\partial\pi_\tau$ (cf. Eq. (5.46)). According to Eq. (5.90), this requires to solve the nonlinear equation (5.87) and to compute the Lyaponov function (5.89) repeatedly. Finally, the gradients of the normal vector constraint (5.47b)-(5.47d) and the second-order gradients of the smoothed eigenvalue constraint $\alpha_{\bar{J}}^{\epsilon_e}(\cdot)$ need to be computed. However, to the author's knowledge, analytical forms of such second-order gradients are still unknown (cf. [174]). In practice therefore, costly finite difference method need to be applied.

For complex reactor networks, the sampling method for treating the SIP (cf. Eq. (5.100)) is applied before applying the normal vector approach. This intermediate step is also computationally extremely demanding, which may take up to several days to finish. This is mainly because the number of sampled points π_τ^k, $k = 1, \cdots, N_{sp}$, in the uncertainty region grows exponentially as the dimension of uncertain parameters. Eq. (5.100) may easily result in a large number of nonlinear smoothened eigenvalue constraints, which are costly to evaluate.

https://doi.org/10.51202/9783186952035 - Generiert durch IP 216.73.216.36, am 18.01.2026, 21:11:22. © Urheberrechtlich geschützter Inhalt. Ohne gesonderte Erlaubnis ist jede urheberrechtliche Nutzung untersagt, insbesondere die Nutzung des Inhalts im Zusammenhang mit, für oder in KI-Systemen, KI-Modellen oder Generativen Sprachmodellen.

Although the solution of the two subproblems derived from the proposed two-step solution strategy is computationally demanding and non-trivial, problem (6.21) can be solved in a very efficient manner. For the reactor network shown in Fig. 6.12, the open-loop Jacobian matrix J has a dimension of 68×68, the candidate CV Δy is a 5-dimensional vector, while the candidate MV is a 15-dimensional vector, refer to Eq. (6.18). The derived optimization problem (6.21) can be solved typically by SNOPT in less than 5 minutes, starting from randomly selected initial points. The convergence behavior of the applied NLP solver is also satisfactory. This observation suggests again that the large computational load of solving problems (3.44) or (4.39) is, most likely, due to the appearance of a large number of nonlinear constraints resulting from the discretization of PFR.

We note that a relatively low number of discretization points is chosen to approximate the PFR in our case studies due to limited computational power. Each PFR is discretized by only 5 points so that a treatable size of the system's Jacobian matrix can be obtained.

https://doi.org/10.51202/9783186952035 - Generiert durch IP 216.73.216.36, am 18.01.2026, 21:11:22. © Urheberrechtlich geschützter Inhalt. Ohne gesonderte Erlaubnis ist jede urheberrechtliche Nutzung untersagt, insbesondere die Nutzung des Inhalts im Zusammenhang mit, für oder in KI-Systemen, KI-Modellen oder Generativen Sprachmodellen.

7 Summary and outlook

7.1 Contributions and summary of this work

In this work, a novel methodology is presented for open-loop and closed-loop reactor network design with guaranteed robust dynamic properties. Both flowsheet and control structural alternatives are parametrized by continuous and discrete design degrees of freedom, which are determined by solving a single optimization problem. The method has been successfully applied to an allyl chloride case study, in which up to 5 PFR and CSTR are included in the reactor network superstructure.

A structured modeling procedure is proposed to formulate the open-loop model of the reactor network. Each reactor in the reactor network superstructure is modeled by a set of differential equations; the connections of the reactor inlets and outlets are modeled by algebraic equations. This modeling procedure results in a dynamic model which covers flowsheet structural information of the open-loop reactor network. In formulating the closed-loop model for simultaneous process and control design, decentralized PI controllers are added to the open-loop reactor network. The paring of candidate CV and MV is not assumed a priori, such that the control structure is not fixed and also subject to optimization. The obtained dynamic model for closed-loop design can therefore determine both flowsheet and control structural alternatives simultaneously.

Idle/non-idle reactors and controllers are allowed to appear in the final design. Idle/non-idle reactors are identified by a zero or non-zero values of the flowrate variables q, which physically represent the valve positions in connecting pipes. Idle/non-idle controllers are identified by the controller gain matrix K, which physically represents the coupling of candidate CV and MV. Both, the flowrate variables and the control gain matrix, are design degrees of freedom. Hence, manipulating their values changes the status of each reactor or controller from idle to non-idle, or vice versa. This trick is central to the formulation and discussion presented in this work, in particular to the analysis of the eigenvalue constraints for reactor network design. A great advantage of considering idle/non-idle reactors and controllers is that the optimal number and type of used reactors and controllers in the final design can be decided.

Eigenvalue-based constraints guarantee the dynamic properties of stability and response speed in this work. The spectral abscissa of the system's Jacobian matrix quantifies the Lyapunov stability and the response speed of an operating point with respect to slow disturbances and control signals. If the spectral abscissa is negative, the system is asymptotically stable. The more negative the spectral abscissa is, the faster the response speed. Constraining the spectral abscissa of the Jacobian matrix by a negative upper bound therefore results in the so-called eigenvalue constraint for the design of dynamic systems with specified dynamic properties.

Although the derived dynamic models for open-loop and closed-loop reactor network design are in the form of ODE systems, we can not directly apply the standard theory of dynamic systems to formulate an eigenvalue constraint for the entire reactor network

https://doi.org/10.51202/9783186952035 - Generiert durch IP 216.73.216.36, am 18.01.2026, 21:11:22. © Urheberrechtlich geschützter Inhalt. Ohne gesonderte Erlaubnis ist jede urheberrechtliche Nutzung untersagt, insbesondere die Nutzung des Inhalts im Zusammenhang mit, für oder in KI-Systemen, KI-Modellen oder Generativen Sprachmodellen.

model. This is because idle reactors or controllers may exist in the final design, though we are interested only in the dynamic properties determined by the submodels of non-idle reactors and controllers. To adapt the eigenvalue constraint for the entire reactor network model to an eigenvalue constraint for the submodels of only non-idle reactors and controllers, structural relationships between idle and non-idle reactors and controllers are analyzed. An important finding in this work reveals that the system's Jacobian matrix has an upper-triangular structure. This finding allows the formulation of a novel eigenvalue constraint which takes the dynamic properties of only non-idle reactor and controllers into account.

Two semi-infinite MINLP with disjunctions, complementarity and robust eigenvalue constraints are formulated for the open-loop and the closed-loop reactor network design problems. In both problem formulations, an economical objective function is maximized subject to the steady-state process model, feasibility constraints and a robust eigenvalue constraint. Integer variables represent the existence of reactors and PI controllers. The number of introduced integers equals the total number of reactors and controllers. Parametric uncertainty may either result from model uncertainties such as reaction kinetic constants or heat transfer coefficients, or from process uncertainties including slow disturbances in load or quality of raw materials. We assume that uncertain parameters are located in an uncertainty region around their nominal values. Complementarity constraints are proposed to select the control structure and disjunctions correspond to the definitions of idle/non-idle reactors and controllers. The robust eigenvalue constraint guarantees robust stability and a specified response speed. For the open-loop case, the derived optimization problem determines the optimal process design variables, operating point and flowsheet structure. For the closed-loop case, the control structure and controller parameters are determined in addition.

Besides modeling and problem formulation, another major challenge encountered in this work is the solution of the derived optimization problems, because no existing solution algorithm or software can be applied readily. After reviewing related optimization problems, we proposed a two-step hybrid method to solve the resulting semi-infinite MINLP. In the first step, we solve an optimization problem, in which all uncertain variables are assumed to be at their nominal values. In the second step, we solve a robust problem, in which all integers are fixed to the results of the first step and parametric uncertainty of uncertain variables is considered. The problem in the first step is solved by established local smooth solvers, which either assume that the eigenvalue constraint is smooth at the local minima or which rely on smoothing techniques for the eigenvalue constraints. A multiple-start strategy is applied to approximate the global minimum of the strongly nonlinear and non-convex optimization problem. The problem in the second step is solved by applying the normal vector approach, which guarantees robust dynamic properties. A great advantage of applying this two-step hybrid method is that, the normal vector approach can be properly initialized using the results of the first step such that good local solutions of the original semi-infinite MINLP can be obtained.

A case study of allyl chloride production is finnally presented, in which both PFR and CSTR are included in the reactor network superstructure. Design results derived by applying the proposed method are presented. Numerical continuation of critical boundaries is performed to illustrate the robustness of guaranteed dynamic properties. The simultaneous design results, computed by the proposed method, are also compared with an established design result derived from the sequential design approach by using simulations. Simulation

https://doi.org/10.51202/9783186952035 - Generiert durch IP 216.73.216.36, am 18.01.2026, 21:11:22. © Urheberrechtlich geschützter Inhalt. Ohne gesonderte Erlaubnis ist jede urheberrechtliche Nutzung untersagt, insbesondere die Nutzung des Inhalts im Zusammenhang mit, für oder in KI-Systemen, KI-Modellen oder Generativen Sprachmodellen.

studies show that the simultaneous design responds significantly faster than the sequential design. This observation demonstrates the power of the proposed method in designing closed-loop reactor networks.

We would like to stress two important modeling tricks, which play a key role in this work. These two tricks render the numerical solution of the formulated optimization problems much more efficient. They also reduce the number of integers included in the problem formulation significantly.

The first trick refers to Eq. (3.4), which assumes that each reactor outlet can be modeled by multiplying a scalar flowrate variable $q_{i,j}$, which belong to the degrees of freedom of the reactor network model, with a vector $g_{i,j}(\cdot)$. This way, one can cut off the connections between reactors by simply manipulating the continuous design degrees of freedom q in the reactor network model. This useful property leads directly to a structured Jacobian matrix J_{tot} with diagonal submatrices for the open-loop reactor network, refer to in Eq. (3.32). The discovery of the inner structure of the Jacobian matrix J_{tot} leads to the important conclusion that the spectral abscissa of the Jacobian matrix with respect to only non-idle reactors is discontinuous.

The second trick refers to Eq. (4.11), which models decentralized control structure alternatives, i.e., the different ways of pairing candidate CV and MV, by using complementarity constraints. Compared with previous suggested integer-based formulation [124], our formulation does not include any integer variables to model decentralized control structure alternatives. Since complementarity constraints can be treated more efficiently than integer variables by local optimization solvers, the novel formulation significantly reduces the computational demand. Therefore, this formulation makes it possible to handle control structure selection problems with a relatively larger number of candidate CV and MV. According to our computational experiences, the method works for control structure selection problems with tens of candidate CV and MV. The formulation has also robust numerical performance and it has potential to be applied to other control design problems.

7.2 Future research directions

7.2.1 Extensions regarding to the guaranteed robust dynamic properties

In the proposed problem formulations (3.44) and (4.39), only the spectral abscissa of an analyzed nonlinear system is considered as a design criterion, which guarantees stability and provides bounds on the decay rate. However, we need to stress that the proposed method assumes that the nonlinear system can be approximated well by linearization. If this is not the case, the actual response of the studied nonlinear system may deviate largely from its linearized system, easily leading to stability loss or unexpected dynamic behavior. Therefore, the proposed method in this work may result in unsatisfactory designs. To avoid this problem, one may need to consider for example the transient solution or other design criteria of nonlinear dynamic systems as well.

The proposed method guarantees not only dynamic properties at the nominal operating point, but also dynamic properties in the uncertainty region. However, the presented formulations are restricted to treat parametric uncertainties, e.g., reaction kinetic parameters, or quantities that can be modeled by parametric uncertainty, e.g., slow disturbances

https://doi.org/10.51202/9783186952035 - Generiert durch IP 216.73.216.36, am 18.01.2026, 21:11:22. © Urheberrechtlich geschützter Inhalt. Ohne gesonderte Erlaubnis ist jede urheberrechtliche Nutzung untersagt, insbesondere die Nutzung des Inhalts im Zusammenhang mit, für oder in KI-Systemen, KI-Modellen oder Generativen Sprachmodellen.

in the input variables, disturbances and reference signals. Fast disturbances, which are often present in chemical processes, have not been considered in this work. To model slow disturbances as parametric uncertainties, we assumed that they can be partitioned into a mean value and a bounded time-dependent variation, which varies *quasi-statically* compared to the system dynamics (cf. [119]). Eqs. (3.44c), (4.39f) are derived as a consequence of this assumption. They ensure the satisfaction of the specified dynamic properties for all uncertain parameters π_τ locating in the uncertainty region. To extend the current work to consider fast disturbances in the future, one may need to consider the non-steady state transient behavior for the robust design of nonlinear systems presented in [53].

7.2.2 Extensions regarding optimization methods

Properly solving problem (3.44) and (4.39) to local and global minima remains an unsolved task in mathematical programming. It is currently very difficult even to get local minima reliably and extremely challenging to get the global one. The proposed two-step hybrid solution method in Section 5.6 may result in sub-optimal solutions and the solution method may fail, if for example the uncertainty region is too large. If one aims to solve more complicated design problems of reactor network synthesis, in the author's opinion, properly solving the optimization problems in the form of Eq. (4.39) is one of the most urgent tasks.

One of the most difficult challenges of solving problem (3.44) and (4.39) is to reliably solve the eigenvalue optimization (EVO) problems to local and global optimality. This is, however, not trivial, since developing local solution methods for EVO problems is still an active research field (cf. Section 5.4). The eigenvalue constraint is non-Lipschitz continuous (cf. Section 2.2) and the state-of-the-art results of non-smooth optimization techniques assume Lipschitz continuity (cf. Section 5.4.2 and [8]). For non-Lipschitz functions, its gradient information is not defined mathematically, and it is not covered by the generalized subdifferential or subgradient (cf. Definition 5.4.2). Without properly defining the gradient information of eigenvalue constraints, it is almost impossible to formulate the local optimality condition of EVO problems. Due to this issue, the existing non-smooth optimization algorithms can not be directly applied either. Global solutions of EVO or non-smooth optimization problems, to the author's knowledge, are rarely addressed in the literature.

Another major challenge is the solution of the semi-infinite optimization (SIP) embedded with a non-smooth inner EVO problem. Because for SIP global optimality of the inner optimization problem is required, the inner EVO problem must be solved to global optimality, which is a challenge task. If the local reduction method or the discretization (cf. Section 5.3.1) is going to be applied, convergence must be proven for SIP with a non-smooth inner optimization problem. A practical way to do this is to apply the smoothing techniques of eigenvalue constraints (cf. Section 5.4.3), which results in a SIP with a smooth inner NLP. Global solution of the smoothened eigenvalue problems should be addressed in this case first. Note that the situation of solving problem (3.44) and (4.39) will get more complicated, if complementarity constraints and disjunctions are included in addition.

For the applied NVA, the local convergence property is rigorously proved in Theorem 5.3.10 for the considered type of SIP (5.42). Global convergence of the NVA should be established afterwards. Local convergence ensures that the developed algorithm can converge to a local minimum of the original SIP, if an initial point is selected which is sufficiently

https://doi.org/10.51202/9783186952035 - Generiert durch IP 216.73.216.36, am 18.01.2026, 21:11:22. © Urheberrechtlich geschützter Inhalt. Ohne gesonderte Erlaubnis ist jede urheberrechtliche Nutzung untersagt, insbesondere die Nutzung des Inhalts im Zusammenhang mit, für oder in KI-Systemen, KI-Modellen oder Generativen Sprachmodellen.

close to a local minimum. Global convergence ensures that the convergence is independent of the selection of the initial point and therefore remote initial points can be selected (cf. also the local and global convergence issue of the local reduction method presented in Section 5.3.1). In the original work on the NVA [119], the authors provided a mechanism of detecting and updating critical boundaries, which relates to the global convergence issue of the NVA. However, in the author's opinion, the global convergence property of this mechanism has not been proven rigorously. To work on this topic, one may start again from analyzing the simplified SIP (5.42), in which all constraints are smooth and no equality constraints are present, and try to establish global convergence of the NVA. The penalty function (5.33), the mechanism of detecting critical manifolds (namely updating the index set K in Eq. (5.32) or J in problem (5.68)) and the selection of step length in Eq. (5.34), which are developed originally for the local reduction method of general SIP, may be reused and adapted.

7.2.3 Extension to control structure selection for linear system

In this work, we have presented a problem formulation (4.39) for the simultaneous process and control system design problem. According to our computational experience and the previous discussion, we find that the presented problem is still too complicated to be solved. Even the local minima can not be obtained properly and getting the global minima is extremely challenging. Motivated by this fact, we present here an interesting problem formulation for control structure selection for linear dynamic systems. The formulation is based on the proposed complementarity constraints (4.11) and it aims to determine an optimal decentralized control structure, including the controller gain parameters, such that a given cost function is minimized. Compared with problem (4.39), this problem is much simpler and we expect that there is a good chance to solve it efficiently.

The control structure selection problem for linear dynamic systems can be formulated as

$$\begin{aligned}
&\min_{K_v, K_v^+, K_v^-, T_v, p} \quad \varphi(x(t), e(t), u(t), y(t), K_v, T_v, p) && \text{(7.1a)}\\
&\qquad s.t.\ \dot{x} = A(p)x + B(p)u + C(p)d_0(t), && \text{(7.1b)}\\
&\qquad\quad y = D(p)x + E(p)u + F(p)d_0(t), && \text{(7.1c)}\\
&\qquad\quad \dot{e} = y, && \text{(7.1d)}\\
&\qquad\quad u = K(y + Te), && \text{(7.1e)}\\
&\qquad\quad [K]_{v,w} = [K^+]_{v,w} - [K^-]_{v,w}, \quad v = 1, ..., n_m, w = 1, ..., n_c, && \text{(7.1f)}\\
&\qquad\quad [\hat{K}]_{v,w} = [K^+]_{v,w} + [K^-]_{v,w}, \quad v = 1, ..., n_m, w = 1, ..., n_c, && \text{(7.1g)}\\
&\qquad\quad 0 \le [K^+]_{v,w} \perp [K^-]_{v,w} \ge 0, \quad v = 1, ..., n_m, w = 1, ..., n_c, && \text{(7.1h)}\\
&\qquad\quad 0 \le [\hat{K}]_{v,w} \perp \sum_{w' \neq w} [\hat{K}]_{v,w'} \ge 0, \quad v = 1, ..., n_m, w = 1, ..., n_c, && \text{(7.1i)}\\
&\qquad\quad 0 \le [\hat{K}]_{v,w} \perp \sum_{v' \neq v} [\hat{K}]_{v',w} \ge 0, \quad v = 1, ..., n_m, w = 1, ..., n_c, && \text{(7.1j)}\\
&\qquad\quad x(0) = x_0,\ e(0) = e_0,\ t \in [0, t_f], && \text{(7.1k)}
\end{aligned}$$

https://doi.org/10.51202/9783186952035 - Generiert durch IP 216.73.216.36, am 18.01.2026, 21:11:22. © Urheberrechtlich geschützter Inhalt. Ohne gesonderte Erlaubnis ist jede urheberrechtliche Nutzung untersagt, insbesondere die Nutzung des Inhalts im Zusammenhang mit, für oder in KI-Systemen, KI-Modellen oder Generativen Sprachmodellen.

where $x(t) \in \mathbb{R}^{n_x}$, $y(t) \in \mathbb{R}^{n_c}$, $u(t) \in \mathbb{R}^{n_m}$ denote the state variables, the candidate CV and the candidate MV of linear system (7.1b), (7.1c). $p \in \mathbb{R}^{n_p}$ denote the design parameters. $d_0(t)$ denote typical disturbances, which are assumed to be known a priori. These disturbances may either be slow disturbances, whose time scales are considerably slower than the time scales of the process (low frequency disturbances), or fast disturbances, whose time scales are similar to the time scales of the process (high frequency disturbances).

$A(p)$, $\cdots$, $F(p)$ are matrices with proper dimensions, which depend on parameter p. $e \in \mathbb{R}^{n_c}$ denote the state variables of n_c candidate PI controllers. $K \in \mathbb{R}^{n_m \times n_c}$ is the proportional control gain matrix. $T := diag(1/t_1, \cdots, 1/t_{n_c}) \in \mathbb{R}^{n_c \times n_c}$ is the integral control gain matrix, where $t_i \in \mathbb{R}$, $i = 1, \cdots, n_c$, refers to the integral control gain of the i-th controllers (cf. Eq. (4.6)). $[K]_{v,w}$, $[K^+]_{v,w}$ and $[K^-]_{v,w}$ denote the (v,w)-th element of matrices K, K^+ and K^- (cf. Eq. (4.11)). K_v, K_v^+, K_v^- and T_v concatenate the variables in matrices K, K^+, K^- and T, respectively. x_0 and e_0 denote initial conditions. t_f denotes the time period over which the optimization is carried out.

Eqs. (7.1b)-(7.1e) refer to a classical multi-input multi-output closed-loop linear system. Eqs. (7.1b), (7.1c) refer to the open-loop system, while Eqs. (7.1d), (7.1e) refer to the state equations of n_c PI controllers and the coupling of candidate CV and MV (not necessarily decentralized). Eqs. (7.1f)-(7.1j) are reproduced from Eq. (4.11), which ensure that each row and each column of matrix K have at most one non-zero element. This way, a decentralized PI control structure can be guaranteed. Eq. (7.1a) is a generalized objective function, which is dependent on the transient solution of the closed-loop system (7.1b)-(7.1e) for $t \in [0, t_f]$.

Problem (7.1) is a dynamic optimization problem with complementarity constraints (but without integer variables). Dynamic optimization refers to mathematical programs in which the objective function and constraints depend on the solution of differential equations. By solving problem (7.1), we aim to find an optimal decentralized PI control structure (embedded in the optimal solution of matrix K), integral control gain matrix T and design parameters p such that the objective function $\varphi(\cdot)$ is minimized, when disturbances $d_0(t)$ are present to the system. Note that one does not have to use all candidate CV or MV to form closed-loops. Solving problem (7.1) also determines how many PI controllers should be included in the final design.

The solution strategies for dynamic optimization problems can be classified into indirect and direct methods. Binder et al. [20] provide a literature review of this topic and a comparison between the indirect and direct methods. Indirect methods require knowledge on the structure of the optimal control profiles in order to derive the boundary value problem representing the first-order necessary conditions of optimality [159]. They often results in a two-point boundary value problem and this may be as difficult to solve as the original optimization problem [191]. In case of nonlinear path constrained problems involving states and controls, as they are frequently encountered in chemical engineering applications, the solution structure is not known a priori [7]. Thus, the application of the indirect approaches is rather cumbersome or even impossible [20]. Direct methods rely on a discretization of the optimal control problem and apply nonlinear programming techniques to solve the resulting finite-dimensional NLP. There are two main approaches to the discretization, the sequential approach in which only the decision variables (control parameters) are discretized and the simultaneous approach, in which both decision and state variables are discretized [191]. Direct approaches have been proven to efficiently solve large-scale optimal control and nonlinear model predictive control problems [61].

https://doi.org/10.51202/9783186952035 - Generiert durch IP 216.73.216.36, am 18.01.2026, 21:11:22. © Urheberrechtlich geschützter Inhalt. Ohne gesonderte Erlaubnis ist jede urheberrechtliche Nutzung untersagt, insbesondere die Nutzung des Inhalts im Zusammenhang mit, für oder in KI-Systemen, KI-Modellen oder Generativen Sprachmodellen.

Future works should focus on solving problem (7.1) to local or even global optimality. For global solutions of dynamic optimization problems we refer to the works of [103, 158, 191] and the references therein. To solve problem (7.1) properly, however, one should also take care of the complementarity constraints (7.1f)-(7.1j). For a first investigation, one may consider using the regularization methods presented in Section 5.2.2. Computational performances of solving (7.1) and solving the control structure selection problem based on integer variables [124] can be compared. Solving problem (7.1) efficiently and reliably is an interesting task for control system design.

7.2.4 Other possible extensions and improvements

In this work, we use the big-M formulations (3.42) and (4.36) to reformulate the discontinuous eigenvalue constraints (3.33) and (4.35) into continuous ones. There may exist, however, other ways to treat the numerical difficulty caused by discontinuity. The current big-M formulations have the drawback that in practice the value of M can not be chosen arbitrarily large, because it will make the derived eigenvalue constraints ill-conditioned. A tight estimation of M is therefore needed. However, this is not a trivial task, because the estimation of the smallest M is itself a global eigenvalue optimization problem.

The presented case study considers PFR in the reactor network superstructure. However, because of limited computational capacity, each PFR is discretized only by 5 points along the tube (One may also consider that each PFR is approximated by 5 CSTR.). This way, the calculated results may not be sufficiently accurate. To have more accurate results, one may need a finer discretization. Future work should readdress this issue, if an efficient numerical optimizer becomes available.

Reaction-separation-recycle is a typical flowsheet structure of chemical processes, which is frequently used in practice. This work focuses only on the reaction part and neglects the effects caused by the separation part and the recycles. It is an interesting task, if one can include distillation columns and recycles into the proposed design framework. This way, robust designs of entire process flowsheets with guaranteed dynamic properties can be addressed. However, this problem is computationally much more complicated than the current problem. We would therefore not suggest to look at this problem in the near future, but suggest to first develop reliable and efficient numerical solvers for the current problem of reactor network synthesis or even simplified ones.

Complementarity constraints can be used to offer an alternative way of formulating logic and discrete relationships, which are traditionally done by using integer variables and disjunctions (cf. Section 5.1). For example, the complementarity constraint (4.9) is an alternative representation of

$$\begin{aligned} &\begin{bmatrix} z_i \\ [\xi]_i > 0 \end{bmatrix} \vee \begin{bmatrix} \bar{z}_i \\ [\xi]_i = 0 \end{bmatrix}, \ i = 1, \cdots, n_\xi, \\ &\sum_{i=1,\cdots,n_\xi} z_i \leq 1, \\ &z_i \in \{0, 1\}, \end{aligned} \tag{7.2}$$

where $[\xi]_i$ denotes the i-th element of vector $\xi \in \mathbb{R}^{n_\xi}$ (cf. Eq. (4.9)). z_i is an integer variable, representing whether $[\xi]_i$ is positive or zero. Eq. (7.2) ensures that at most one element of ξ can be positive, which is equivalent to the relationship ensured by Eq.

https://doi.org/10.51202/9783186952035 - Generiert durch IP 216.73.216.36, am 18.01.2026, 21:11:22. © Urheberrechtlich geschützter Inhalt. Ohne gesonderte Erlaubnis ist jede urheberrechtliche Nutzung untersagt, insbesondere die Nutzung des Inhalts im Zusammenhang mit, für oder in KI-Systemen, KI-Modellen oder Generativen Sprachmodellen.

(4.9). In other words, the disjunctions and the integer constraints (7.2) can be represented equivalently by the complementarity constraints (4.9).

Motivated by this example, there may exist other mixed-integer constraints and/or disjunctions, which can be formulated equivalently by complementarity constraints. There exists already some related work [14] [15] [16]. However, a systematic framework to automatically transform mixed-integer constraints and disjunctions into complementarity constraints has not been rigorously proposed. Because the logics and relationships formulated by complementarity constraints are not as straightforward as the ones formulated by disjunctions and integer constraints, it is more convenient for people to start modeling using disjunctions and integer constraints, and then transform them into complementarity constraints for numerical computation. If this is the case, computational performance of complementary-based formulations should be compared with the formulations based on disjunctions and integer constraints.

https://doi.org/10.51202/9783186952035 - Generiert durch IP 216.73.216.36, am 18.01.2026, 21:11:22. © Urheberrechtlich geschützter Inhalt. Ohne gesonderte
Erlaubnis ist jede urheberrechtliche Nutzung untersagt, insbesondere die Nutzung des Inhalts im Zusammenhang mit, für oder in KI-Systemen, KI-Modellen oder Generativen Sprachmodellen.

Appendices

https://doi.org/10.51202/9783186952035 - Generiert durch IP 216.73.216.36, am 18.01.2026, 21:11:22. © Urheberrechtlich geschützter Inhalt. Ohne gesonderte
Erlaubnis ist jede urheberrechtliche Nutzung untersagt, insbesondere die Nutzung des Inhalts im Zusammenhang mit, für oder in KI-Systemen, KI-Modellen oder Generativen Sprachmodellen.

A Extension of the reactor network model to PFR

Plug flow reactors (PFR) represent tubular reactors, where the velocity of the fluid is assumed to be constant across any cross-section perpendicular to the flow direction in the tube. Here, we demonstrate how the partial differential equation (PDE) model of a PFR can be transcribed into an ODE model by the Method of Lines [151]. After this transcription, the PFR model is the same as Eqs. (3.3), (3.4), the models of the reactors in the network. Hence, the proposed synthesis method can be applied straightforwardly to reactor network superstructures including CSTR and PFR.

Assume that the i-th reactor in a reactor network superstructure, Fig. 3.1, is a PFR. Denote L_i as the length of the PFR and denote $\mu' \in [0, L_i]$ as the axial coordinate of the PFR. L_i is the length of PFR i. Let us firstly scale $\mu' \in [0, L_i]$ into $[0, 1]$ for convenience by

$$\mu = \mu'/L_i, \mu \in [0, 1]. \tag{A.1}$$

PFR i can be modeled by

$$\frac{\partial x_i}{\partial t} = f_i(\frac{\partial^2 x_i}{\partial \mu^2}, \frac{\partial x_i}{\partial \mu}, q_{i,1}, \cdots, q_{i,N}, p_i), \tag{A.2}$$

with initial conditions

$$x_i(\mu, 0) = x_{i0}(\mu), \tag{A.3}$$

and boundary conditions

$$x_i(0, t) = w(u_{i,1}(t), \cdots, u_{i,N}(t), q_{i,1}, \cdots, q_{i,N}, p_i), \tag{A.4a}$$

$$\frac{\partial x_i}{\partial \mu}(1, t) = 0. \tag{A.4b}$$

The definitions and physical units of x_i and $u_{i,k}$, $k = 1, \cdots, N$, are the same as in Eq. (3.3). However, the state variables $x_i(\mu, t)$ is not only a function of time, but also a function of axial position μ. $q_{i,j}$, $j = 1, \cdots, N$, denote volumetric flowrate in $[m^3/s]$ through outlet port (i, j) of the PFR and p_i denotes design parameters of PFR i, including the length L_i of the reactor, its cross section, etc. Input variables $u_{i,k}$ enter the left boundary conditions through a smooth function $w(\cdot)$.

Each PFR has N outlets, which are fed into other reactors in the network. The j-th outlet of the PFR can be represented by

$$y_{i,j}(t) = q_{i,j} g_{i,j}(x_i(1, t), p_i), \tag{A.5}$$

where $y_{i,j}$ is of the same type as in Eq. (3.4).

https://doi.org/10.51202/9783186952035 - Generiert durch IP 216.73.216.36, am 18.01.2026, 21:11:22. © Urheberrechtlich geschützter Inhalt. Ohne gesonderte Erlaubnis ist jede urheberrechtliche Nutzung untersagt, insbesondere die Nutzung des Inhalts im Zusammenhang mit, für oder in KI-Systemen, KI-Modellen oder Generativen Sprachmodellen.

We use the methods of lines to transcribe the PDE system (A.2)-(A.4) into an ODE system. The function $x_i(\mu, t)$ is discretized in the spatial coordinate μ in $[0, 1]$, such that the discretized solution only depends on time. $N_d + 2$ positions $\mu_0, \cdots, \mu_{N_d+1}$ are selected along the spatial $[0, 1]$ with $\mu_0 = 0$ and $\mu_{N_d+1} = 1$. Thus, we can introduce

$$x_i^s(t) = x_i(\mu_s, t),\ s = 0, \cdots, N_d + 1,$$

to approximate $x_i(\mu, t)$ by the set

$$\{x_i^0(t), \cdots, x_i^{N_d+1}(t)\}.$$

In this work, we use central differences to approximate derivatives $\partial x_i/\partial\mu$ and $\partial^2 x_i/\partial\mu^2$ to first order accuracy at points $s = 1, \cdots, N_d$ to result in the ODE system

$$\begin{aligned} \dot{x}_i^1(t) &= f_i(x_i^0, x_i^1, x_i^2, q_{i,1}, \cdots, q_{i,N}, p_i),\ x_i^1(0) = x_{i0}^1, \\ &\vdots \\ \dot{x}_i^{N_d}(t) &= f_i(x_i^{N_d-1}, x_i^{N_d}, x_i^{N_d+1}, q_{i,1}, \cdots, q_{i,N}, p_i),\ x_i^{N_d}(0) = x_{i0}^{N_d}. \end{aligned} \tag{A.6}$$

Note that Eqs. (A.6) do not include state equations for $\dot{x}_i^0$ and $\dot{x}_i^{N_d+1}$, because they are already fixed by the boundary conditions Eq. (A.4). With Eq. (A.4), we replace x_i^0 and $x_i^{N_d+1}$ in Eq. (A.6) by

$$x_i^0 = x_i(0, t) = w(u_{i,1}(t), \cdots, u_{i,N}(t), q_{i,1}, \cdots, q_{i,N}, p_i),$$

and

$$0 = \frac{\partial x_i}{\partial\mu}(1, t) \approx \frac{x_i^{N_d+1}(t) - x_i^{N_d}(t)}{\Delta\mu}.$$

The resulting set of N_d ODE is

$$\dot{x}_i^s = f(x_i, u_{i,1}, \cdots, u_{i,N}, q_{i,1}, \cdots, q_{i,N}, p_i),\ x_i^s(0) = x_{i0}^s,\ \forall s = 1, \cdots, N_d. \tag{A.7}$$

After discretization, Eq. (A.5) can be written as

$$y_{i,j}(t) = q_{i,j} g_{i,j}(x_i^{N_d}(t), p_i). \tag{A.8}$$

If we use $x_i := (x_i^{1T}, \cdots, x_i^{NT})^T$ to denote the state variables, Eq. (A.7) has exactly the same form as Eq. (3.3), and Eq. (A.8) has the same form as Eq. (3.4).

Example (continued, refer to Example 3.1 in Chapter 3)**.** *Consider that the i-th reactor is a PFR. The mass balance of component A is*

$$\frac{\partial c_{iA}(\mu', t)}{\partial t} = k_d \frac{\partial^2 c_{iA}(\mu', t)}{\partial\mu'^2} - \frac{1}{S_i}\sum_{j=1}^{N} q_{i,j} \cdot \frac{\partial c_{iA}(\mu', t)}{\partial\mu'} + R_A, \tag{A.9}$$

where S_i denotes the cross section area in $[m^2]$, L_i the length of the reactor in $[m]$. k_d is a mass dispersion coefficient.

Initial and boundary conditions are

$$c_{iA}(\mu', 0) = c_{iA0}(\mu'), \tag{A.10}$$

https://doi.org/10.51202/9783186952035 - Generiert durch IP 216.73.216.36, am 18.01.2026, 21:11:22. © Urheberrechtlich geschützter Inhalt. Ohne gesonderte Erlaubnis ist jede urheberrechtliche Nutzung untersagt, insbesondere die Nutzung des Inhalts im Zusammenhang mit, für oder in KI-Systemen, KI-Modellen oder Generativen Sprachmodellen.

$$\begin{aligned} c_{iA}(0,t) &= \frac{\sum\limits_{k=1}^{N} \dot{n}^0_{Ai,k}}{\sum\limits_{j=1}^{N} q_{i,k}}, \\ \frac{\partial c_{iA}}{\partial \mu'}(L_i,t) &= 0. \end{aligned} \tag{A.11}$$

$\dot{n}^0_{Ai,k}$ $[mol/s]$ denotes the molar flowrate of A through inlet port (i,k) and $\sum\limits_{j=1}^{N} q_{i,k}$ $[m^3/s]$ the volumetric flowrate in the PFR. The mass balances for B and C are set up in the same way.

The energy balance is

$$\begin{aligned} c_p \frac{\partial T_i(\mu',t)}{\partial t} =& k_c \frac{\partial^2 T_i(\mu',t)}{\partial \mu'^2} - \frac{1}{S_i} \sum_{j=1}^{N} q_{i,j} \cdot \frac{c_p \partial T_i(\mu',t)}{\partial \mu'} \\ &+ \sum_{j=1,2,3} H_j r_j + Q_h/S_i/L_i, \end{aligned} \tag{A.12}$$

with the heat capacity c_p $[J/m^3/K]$, the energy dispersion coefficient k_c $[J/K/m/s]$, the heat exchange rate with the reactor jacket Q_h $[J/s]$. The heats of reaction H_j are already defined in Table 3.1.

Initial and boundary conditions for energy balance are

$$T_i(\mu',0) = T_{i0}(\mu'), \tag{A.13}$$

$$\begin{aligned} T_i(0,t) &= \frac{\sum\limits_{k=1}^{N} \dot{Q}^0_{i,k}}{c_p \sum\limits_{j=1}^{N} q_{i,k}}, \\ \frac{\partial T_i}{\partial \mu'}(L_i,t) &= 0, \end{aligned} \tag{A.14}$$

with energy flowrate $\dot{Q}^0_{i,k}$ $[J/s]$ entering the reactor through inlet port (i,k).

The output variables $y_{i,j}$ are

$$y_{i,j}(t) = q_{i,j}(c_A(L_i,t), c_B(L_i,t), c_C(L_i,t), c_p T(L_i,t))^T. \tag{A.15}$$

If we denote

$$\begin{aligned} x_i &= (c_{Ai}, c_{Bi}, c_{Ci}, T_i)^T, \\ u_{i,j} &= (\dot{n}^0_{Ai,j}, \dot{n}^0_{Bi,j}, \dot{n}^0_{Ci,j}, \dot{Q}^0_{i,j})^T, \end{aligned} \tag{A.16}$$

and use Eq. (A.1) to scale the position variables μ', the PDE model (A.9)-(A.15) is exactly the same as Eqs.(A.2)-(A.5). ☐

https://doi.org/10.51202/9783186952035 - Generiert durch IP 216.73.216.36, am 18.01.2026, 21:11:22. © Urheberrechtlich geschützter Inhalt. Ohne gesonderte Erlaubnis ist jede urheberrechtliche Nutzung untersagt, insbesondere die Nutzung des Inhalts im Zusammenhang mit, für oder in KI-Systemen, KI-Modellen oder Generativen Sprachmodellen.

B Proof of Proposition 3.2.1

Proof. Here, we prove that Eq. (3.34) is a sufficient condition for the continuity of $\alpha_{J_{nid}}(\cdot)$. Denote $y = (x^T, q^T, p^T, p_{sys}^T)^T$ and denote U_{y^*} as a neighborhood of y^*. Because in a sufficiently small neighborhood U_{y^*} any non-zero flowrate variables evaluated at point y^* (elements of q in vector y) remain non-zero inside U_{y^*} and only zero flowrate variables evaluated at point y^* may become non-zero inside U_{y^*}, we have

$$\mathcal{I}_{nid}(y') \supseteq \mathcal{I}_{nid}(y^*),\ \forall y' \in U_{y^*}. \tag{B.1}$$

In other words, the size of matrix $J_{nid}(y')$ evaluated at y' is not smaller than the size of matrix $J_{nid}(y^*)$ at y^*.

At point y^*, without loss of generality, assume that reactors $1, \cdots, \theta^*$ are idle, while reactors $\theta^* + 1, \cdots, N$ are non-idle. Then in $U_{y^*}/\{y^*\}$ there exist maximal $N^* = 2^{\theta^*}$ ways to activate the θ^* idle reactors. For example, no idle reactor is activated, only reactor i, $i \in \{1, \cdots, \theta^*\}$, is activated or some reactors of the θ^* idle reactors are activated. Thus, we can denote

$$U_{y^*}/\{y^*\} = \Omega^1 \cup \cdots \cup \Omega^N, \tag{B.2}$$

where Ω^k, $k = 1, \cdots, N^*$, containing points y, refer to the different ways of activation and in each Ω^k the modes (idle or non-idle) of each reactors do not change. Ω^k has the property that $\Omega^{k_1} \neq \Omega^{k_2}$, if $k_1 \neq k_2$. Note that, this partition of $U_{y^*}/\{y^*\}$ by using Ω^k shown in Eq. (B.2) is always valid, for any arbitrarily small neighborhood U_{y^*}. Note also that, each Ω^k does not contain point y^*.

At y^*, denote

$$J_{id}(y^*) = \begin{bmatrix} A_1^* & & 0 \\ & \ddots & \\ 0 & & A_{\theta^*}^* \end{bmatrix}, \tag{B.3}$$

where $A_i^* \in \mathcal{M}_{n_{x_i}}$, $i = 1, \cdots, \theta^*$, refers to the Jacobian matrix of idle reactor i. $\mathcal{M}_{n_{x_i}}$ denote the vector space of n_{x_i}-by-n_{x_i} real matrix. The inner structure of $J_{id}(y^*)$ with diagonal block submatrices follows from the idle-reactor model (3.30).

Consider any $k' \in \{1, \cdots, N^*\}$ and set $\Omega^{k'}$. From the definition of $\Omega^{k'}$, we know that the index set $\mathcal{I}_{nid}(y)$ does not change, for all $y \in \Omega^{k'}$. So we can denote that

$$\mathcal{I}_{nid}(y) \equiv \mathcal{I}_{nid}^{k'}, \forall y \in \Omega^{k'}. \tag{B.4}$$

Without loss of generality, we assume that in $\Omega^{k'}$, reactors $1, \cdots, \theta^* - m^*$ remain idle and reactors $\theta^* - m^* + 1, \cdots, \theta^*$ become non-idle.

Because of (B.4), $\forall y \in \Omega^{k'}$, $J_{nid}(y)$ has a fixed dimension[1] and it can be determined from the non-idle reactor model (3.31) analytically. That is, $\forall y \in \Omega^{k'}$,

$$J_{nid}(y) = [\frac{\partial f_i}{\partial x_j}] \in \mathcal{M}_{l^*},\ i, j \in \mathcal{I}_{nid}^{k'}. \tag{B.5}$$

---

[1] Note that, $J_{nid}(y)$, however, may change size for $y \in U_{y^*}$.

https://doi.org/10.51202/9783186952035 - Generiert durch IP 216.73.216.36, am 18.01.2026, 21:11:22. © Urheberrechtlich geschützter Inhalt. Ohne gesonderte Erlaubnis ist jede urheberrechtliche Nutzung untersagt, insbesondere die Nutzung des Inhalts im Zusammenhang mit, für oder in KI-Systemen, KI-Modellen oder Generativen Sprachmodellen.

l^* is a constant, not depending on different $y \in \Omega^{k'}$.

Because the right hand side of Eq. (B.5) is a smooth function of y, we can take the limit of it, which results in

$$J_{nid}(y) \to \underbrace{\begin{bmatrix} A^*_{\theta^*-m^*+1} & & & 0 \\ & \ddots & & \\ & & A^*_{\theta^*} & \\ 0 & & & J_{nid}(y^*) \end{bmatrix}}_{:=B^*}, \text{as } y \to y^* \text{ and } y \in \Omega^{k'}. \tag{B.6}$$

Furthermore, because of Eq. (B.3) and Eq. (3.34), we have

$$\alpha_{J_{nid}}(y^*) = \alpha_{B^*}.$$

So, applying Lemma 2.2.1 to Eq. (B.6) results in

$$\alpha_{J_{nid}}(y) \to \alpha_{B^*} = \alpha_{J_{nid}}(y^*), \text{as } y \to y^* \text{ and } y \in \Omega^{k'}.$$

In other words, $\forall \epsilon > 0$, there exist a $\delta^{k'} > 0$ so that if $|y - y^*| \leq \delta^{k'}$ and $y \in \Omega^{k'}$, $|\alpha_{J_{nid}}(y) - \alpha_{J_{nid}}(y^*)| < \epsilon$.

Now, define

$$\delta^* = \min_{k'=1,\cdots,N^*} \delta^{k'};$$

because N^* is a finite number, we have $\delta^* > 0$. So if $|y - y^*| \leq \delta^*$, y either equals to y^* or belongs to a set $\Omega^{k'}$, $k' \in \{1, \cdots, N\}$. In both cases, $|\alpha_{J_{nid}}(y) - \alpha_{J_{nid}}(y^*)| < \epsilon$.

Next, we will prove that Eq. (3.34) is a necessary condition of the continuity of $\alpha_{J_{nid}}(\cdot)$. Assume that at y^*, Eq. (3.34) is not satisfied, i.e.

$$\alpha_{J_{id}}(y^*) > \alpha_{J_{nid}}(y^*). \tag{B.7}$$

Construct the following sequences

$$\begin{aligned} q_k &:= q^* + \epsilon_k, \\ y_k &:= (x^{*T}, q_k^T, p^{*T}, p_{sys}^{*T})^T, \end{aligned}$$

where $\epsilon_k > 0 \in \mathbb{R}^{n_q}$, $\epsilon_k \to 0$, and hence, y_k is a sequence approaching y^*. Also, for any q^*, q_k can be selected so that all elements in q_k are not equal to zero, $\forall k = 1, \cdots, \infty$. Hence, all reactors evaluated at points y_k, $k = 1, \cdots, \infty$, are non-idle. So

$$J_{nid}(y_k) = J_{tot}(y_k), \ k = 1, \cdots, \infty. \tag{B.8}$$

We will prove that for the constructed sequence y_k with $y_k \to y^*$, $\alpha_{J_{nid}}(y_k) \nrightarrow \alpha_{J_{nid}}(y^*)$.

Because $\alpha_{J_{tot}}(\cdot)$ is a continuous function (Lemma 2.2.1), we have

$$\alpha_{J_{tot}}(y_k) \to \alpha_{J_{tot}}(y^*), \ k \to \infty. \tag{B.9}$$

Also from Eq. (3.32) and Eq. (B.7), we have

$$\alpha_{J_{tot}}(y^*) = max\{\alpha_{J_{id}}(y^*), \alpha_{J_{nid}}(y^*)\} = \alpha_{J_{id}}(y^*). \tag{B.10}$$

https://doi.org/10.51202/9783186952035 - Generiert durch IP 216.73.216.36, am 18.01.2026, 21:11:22. © Urheberrechtlich geschützter Inhalt. Ohne gesonderte Erlaubnis ist jede urheberrechtliche Nutzung untersagt, insbesondere die Nutzung des Inhalts im Zusammenhang mit, für oder in KI-Systemen, KI-Modellen oder Generativen Sprachmodellen.

Now, if we replace $\alpha_{J_{tot}}(y^*)$ in Eq. (B.9) by using Eq. (B.10), we obtain

$$\alpha_{J_{tot}}(y_k) \to \alpha_{J_{id}}(y^*),\ k \to \infty. \tag{B.11}$$

Use Eq. (B.8), we have

$$\alpha_{J_{nid}}(y_k) \to \alpha_{J_{id}}(y^*), k \to \infty.$$

From Eq. (B.7), we have $\alpha_{J_{nid}}(y_k) \nrightarrow \alpha_{J_{nid}}(y^*)$. This is in contradiction to the continuity of $\alpha_{J_{nid}}(\cdot)$. □

https://doi.org/10.51202/9783186952035 - Generiert durch IP 216.73.216.36, am 18.01.2026, 21:11:22. © Urheberrechtlich geschützter Inhalt. Ohne gesonderte Erlaubnis ist jede urheberrechtliche Nutzung untersagt, insbesondere die Nutzung des Inhalts im Zusammenhang mit, für oder in KI-Systemen, KI-Modellen oder Generativen Sprachmodellen.

C Proof of Proposition 3.2.2

To prove Proposition 3.2.2, let us first introduce a lemma.

Lemma C.1. *The spectral abscissa of a matrix* $G \in \mathbb{R}^{n\times n}$ *can be estimated by*

$$\alpha_G \leq \max_{j=1,\cdots,n}\{G_{jj} + \sum_{i\neq j}|G_{ij}|\}.$$

This lemma is formulated and proved as Theorem 2 in [87]. Now we can prove Proposition 3.2.2 as follows.

Proof. Let us first consider the trivial case, i.e., at point $y^* := (x^{*T}, q^{*T}, p^{*T}, p_{sys}^{*T}, z^{*T})^T$, $\mathcal{I}_{id}(y^*) = \emptyset$. In this case,

$$\alpha_{\bar{J}}(y^*) = \alpha_{J_{tot}}(y^*) = \alpha_{J_{nid}}(y^*).$$

For the non-trivial case, i.e., at point y^* where $\mathcal{I}_{id}(y^*) \neq \emptyset$. From Eq. (3.32), if $z^* = (z_1^*, \cdots, z_N^*)^T$ satisfies Eq. (3.41), then

$$\bar{J}(y^*) = \begin{bmatrix} J_{id}(y^*) - M\cdot I & 0 \\ 0 & J_{nid}(y^*) \end{bmatrix},$$

where I is an identity matrix with the same dimension as $J_{id}(y^*)$. Because the elements in $J_{tot}(y^*)$ are bounded, elements in $J_{id}(y^*)$ and $J_{nid}(y^*)$ are bounded, too. Hence, $\exists b^* \geq 0$ such that

$$\begin{aligned} |\alpha_{J_{nid}}(y^*)| &\leq b^*, \\ \max_j\{(J_{id}(y^*))_{jj} + \sum_{i\neq j}|(J_{id}(y^*))_{ij}|\} &\leq b^*. \end{aligned} \tag{C.1}$$

Furthermore,

$$\begin{aligned} \alpha_{(J_{id}-M\cdot I)}(y^*) &\leq \max_j\{-M + (J_{id}(y^*))_{jj} + \sum_{i\neq j}|(J_{id}(y^*) - M\cdot I)_{ij}|\} \\ &= \max_j\{-M + (J_{id}(y^*))_{jj} + \sum_{i\neq j}|(J_{id}(y^*))_{ij}|\} \\ &\leq -M + b^*. \end{aligned} \tag{C.2}$$

For $M > 2b^*$, combining Eqs. (C.1), (C.2) results in

$$\alpha_{(J_{id}-M\cdot I)}(y^*) < -b^* \leq \alpha_{J_{nid}}(y^*).$$

Thus, $\alpha_{\bar{J}}(y^*) = max\{\alpha_{(J_{id}-M\cdot I)}(y^*), \alpha_{J_{nid}}(y^*)\} = \alpha_{J_{nid}}(y^*)$. ☐

https://doi.org/10.51202/9783186952035 - Generiert durch IP 216.73.216.36, am 18.01.2026, 21:11:22. © Urheberrechtlich geschützter Inhalt. Ohne gesonderte Erlaubnis ist jede urheberrechtliche Nutzung untersagt, insbesondere die Nutzung des Inhalts im Zusammenhang mit, für oder in KI-Systemen, KI-Modellen oder Generativen Sprachmodellen.

D Parametric optimization problems

For $x \in \mathbb{R}^m$, $v \in \mathbb{R}^n$, denote $P(x)$ as an optimization problem depending on parameter x [40, 41, 75, 76, 79]. $P(x)$ takes the form of

$$\min_{v} \; g(x, v) \tag{D.1a}$$

$$s.t. \; h_i(x, v) = 0, i = 1, \cdots, N, \tag{D.1b}$$

$$l_j(x, v) \geq 0, j = 1, \cdots, M. \tag{D.1c}$$

N and M are fixed integers. Functions $g : \mathbb{R}^m \times \mathbb{R}^n \to \mathbb{R}$, $h_i : \mathbb{R}^m \times \mathbb{R}^n \to \mathbb{R}$, $l_j : \mathbb{R}^m \times \mathbb{R}^n \to \mathbb{R}$ are twice continuously differentiable. For each fixed $x = \bar{x}$, $P(\bar{x})$ minimizes the objective function $g(\bar{x}, v)$ subject to constraints (D.1b)-(D.1c), evaluated at $x = \bar{x}$. We are interested here in the local minima of $P(x)$, when x is subject to variation.

Denote $\mathcal{I} = \{1, \cdots, N\}$ and $\mathcal{J} = \{1, \cdots, M\}$ as the index sets for equality and inequality constraints. Denote

$$\mathcal{A}(x, v) = \{j \in \mathcal{J} \mid l_j(x, v) = 0\} \tag{D.2}$$

as the index set of active inequality constraints, which depends on the evaluation point $(x^T, v^T)^T$. Define

$$L(x, v, \lambda, \mu) = g(x, v) + \sum_{i \in I} \lambda_i h_i(x, v) - \sum_{j \in J} \mu_j l_j(x, v) \tag{D.3}$$

as the Lagrange function of $P(x)$, where $\lambda_i \in \mathbb{R}$, $i \in I$, and $\mu_j \in \mathbb{R}$, $j \in J$. Denote $\lambda = (\lambda_1, ..., \lambda_N)^T \in \mathbb{R}^N$ and $\mu = (\mu_1, ..., \mu_M)^T \in \mathbb{R}^M$ as the Lagrangian multipliers for equality and inequality constraints, respectively.

Definition D.1 (LICQ of $P(\bar{x})$)**.** *For fixed $x = \bar{x}$, linear independence constraint qualification (LICQ) is said to hold for $P(\bar{x})$ at $v = \bar{v}$, if the vectors $\nabla_v h_i(\bar{x}, \bar{v})$, $i \in I$, $\nabla_v l_j(\bar{x}, \bar{v})$, $j \in \mathcal{A}(\bar{x}, \bar{v})$ are linearly independent.*

Theorem D.1 (First-order necessary optimality condition, refer to e.g. Theorem 12.1 in [127])**.** *For fixed $x = \bar{x}$, assume v^* is a local minimum of $P(\bar{x})$ and assume LICQ holds at $v = v^*$. Then, there exist unique $\lambda^* \in \mathbb{R}^N$ and $\mu^* \in \mathbb{R}^M$, such that*

$$0 = \nabla_v L^T(\bar{x}, v^*, \lambda^*, \mu^*), \tag{D.4a}$$

$$0 = h_i(\bar{x}, v^*), \forall i \in \mathcal{I}, \tag{D.4b}$$

$$0 = l_j(\bar{x}, v^*), \forall j \in \mathcal{A}(\bar{x}, v^*), \tag{D.4c}$$

$$0 = \mu_j^*, \forall j \notin \mathcal{A}(\bar{x}, v^*), \tag{D.4d}$$

$$0 \leq l_j(\bar{x}, v^*), \forall j \in \mathcal{J}, \tag{D.4e}$$

$$0 \leq \mu_j^*, \forall j \in \mathcal{J}. \tag{D.4f}$$

Eq. (D.4) is the so-called KKT necessary optimality condition of $P(x)$. Vector $(v^{*T}, \lambda^{*T}, \mu^{*T})^T$ is called a KKT point of $P(\bar{x})$, if it satisfies Eq. (D.4).

https://doi.org/10.51202/9783186952035 - Generiert durch IP 216.73.216.36, am 18.01.2026, 21:11:22. © Urheberrechtlich geschützter Inhalt. Ohne gesonderte Erlaubnis ist jede urheberrechtliche Nutzung untersagt, insbesondere die Nutzung des Inhalts im Zusammenhang mit, für oder in KI-Systemen, KI-Modellen oder Generativen Sprachmodellen.

Definition D.2 (Strict complementarity (SC) condition)**.** *For fixed* $x = \bar{x}$*, assume that* $(v^{*T}, \lambda^{*T}, \mu^{*T})^T$ *is a KKT point of* $P(\bar{x})$*, the strict complementarity (SC) condition is said to hold, if*

$$\mu_j^* > 0,\ \forall j \in \mathcal{A}(\bar{x}, v^*). \tag{D.5}$$

Theorem D.2 (Second-order sufficient conditions (SOSC) for a local isolated minimizing point of $P(\bar{x})$, Lemma 2.1 in [40])**.** *If there exist (Lagrange multipliers) vectors* $\lambda^* \in \mathbb{R}^N$*,* $\mu^* \in \mathbb{R}^M$ *such that the KKT condition (D.4) holds for* $x = \bar{x}$ *and* $v = v^*$*, and further if*

$$s^T \nabla_{vv} L(\bar{x}, v^*, \lambda^*, \mu^*) s > 0,$$

for all $s \neq 0$ *such that*

$$\begin{aligned} &\nabla_v l_j(\bar{x}, v^*) s \geq 0,\ \textit{for all } j,\ \textit{where } l_j(\bar{x}, v^*) = 0, \\ &\nabla_v l_j(\bar{x}, v^*) s = 0,\ \textit{for all } j,\ \textit{where } \mu_j^* > 0, \\ &\nabla_v h_i(\bar{x}, v^*) s = 0,\ i = 1, \cdots, N, \end{aligned} \tag{D.6}$$

then v^* *is a local isolated (locally unique) minimizer of* $P(\bar{x})$

Note that under SC condition, Eq. (D.6) is equivalent to

$$\begin{aligned} &\nabla_v l_j(\bar{x}, v^*) s = 0,\ \forall\, j \in \mathcal{A}(\bar{x}, v^*), \\ &\nabla_v h_i(\bar{x}, v^*) s = 0,\ i = 1, \cdots, N. \end{aligned} \tag{D.7}$$

Theorem D.3 (Local minimizer v^* under second-order sufficient conditions, Theorem 2.1 in [40])**.** *If (i) the SOSC in Theorem D.2 for a local minimum of* $P(\bar{x})$ *holds at* v^* *with associated Lagrange multipliers* λ^* *and* μ^**, (ii) LICQ condition of* $P(\bar{x})$ *holds at* $v = v^*$*, (iii) SC condition of* $P(\bar{x})$ *holds at* $v = v^*$*, then:*

(a) v^* *is a local isolated minimizing point of* $P(\bar{x})$ *and the associated Lagrange multipliers* λ^* *and* μ^* *are unique.*

(b) For x *in a neighborhood of* $\bar{x}$*, there exists a unique once continuously differentiable function* $(v(x), \lambda(x), \mu(x))$ *satisfying the SOSC for a local minimum of problem* $P(x)$ *such that* $v(\bar{x}) = v^*$*,* $\lambda(\bar{x}) = \lambda^*$*,* $\mu(\bar{x}) = \mu^*$*, and, hence,* $v(x)$ *is a locally unique local minimum of* $P(x)$ *with associated unique Lagrange multipliers* $\lambda(x)$ *and* $\mu(x)$*.*

(c) SC condition and LICQ hold at $v(x)$ *for* x *near* $\bar{x}$*.*

https://doi.org/10.51202/9783186952035 - Generiert durch IP 216.73.216.36, am 18.01.2026, 21:11:22. © Urheberrechtlich geschützter Inhalt. Ohne gesonderte Erlaubnis ist jede urheberrechtliche Nutzung untersagt, insbesondere die Nutzung des Inhalts im Zusammenhang mit, für oder in KI-Systemen, KI-Modellen oder Generativen Sprachmodellen.

E Proof of Theorem 5.3.10

The proof of Theorem 5.3.10 is separated into two parts, which are stated in Theorem E.1.6 and Theorem E.2.5. In Section E.1, we present a related parametric problem and prove that the feasible set F of the original SIP (5.42), denoted as $\mathcal{P}$, is locally the same as the feasible set F^m defined by using this introduced parametric problem (cf. Theorem E.1.6). In Section E.2, we prove that the feasible set F^m is locally the same as F^n (cf. Theorem E.2.5). Combining the results, Theorem 5.3.10 can be obtained straightforwardly. Note taht because we are only interested in the feasible set of $\mathcal{P}$, the objective function $f(\cdot)$ of $\mathcal{P}$ is of no interest.

E.1 A related parametric problem

Denote set

$$Y(x,y) = \{t \in \mathbb{R}^m \,|\, g(x+t,y) = 0\},$$

$x \in \mathbb{R}^m$ and $y \in \mathbb{R}^n$ are variables as they are defined before. Let us consider the following parametric problem $\mathcal{I}^t(x,y)$,

$$\min_{t \in Y(x,y)} t^T t. \tag{E.1}$$

Note that since $Y(x,y)$ may not be a compact set, $\mathcal{I}^t(x,y)$ may not attain its minimum. Define

$$F^m = \{z = (x^T, y^T)^T \mid t^T t \geq 1, \; \forall\, t \in Y(x,y)\}. \tag{E.2}$$

Lemma E.1.1. *If $T_a(\bar{z}) \neq \emptyset$ and the TC holds at $z = \bar{z}$, then there exists a neighborhood $V_{\bar{z}}$ such that $Y(x,y) \neq \emptyset$, $\forall z \in V_{\bar{z}}$.*

Proof. It is obvious that if $\bar{t} \in T_a(\bar{z})$, $\bar{t} \in Y(\bar{x},\bar{y})$. Hence, from $T_a(\bar{z}) \neq \emptyset$ follows $Y(\bar{x},\bar{y}) \neq \emptyset$ directly. Select any $\bar{t} \in T_a(\bar{z})$, denote $x' = \bar{x} + \bar{t}$, we have $g(x',\bar{y}) = 0$. From TC, without loss of generality, assume that $\nabla_{x_1} g(x',\bar{y}) \neq 0$, where x_i, $i = 1, \cdots, m$, refers to the i-th element of x. From the IFT, $g(x,y) = 0$ locally determines an at least continuous function $x_1(x_2, \cdots, x_m, y)$ for x_i sufficiently closed to x'_i, $i = 2, \cdots, m$, and y sufficiently closed to $\bar{y}$, and

$$g\big(\,(\,x_1(x_2, \cdots, x_m, y),\, x_2,\, \cdots,\, x_m)^T, y\big) \equiv 0.$$

Therefore, for any (x,y) in the neighborhood $V_{\bar{z}}$, define $t' = (x_1(x_2, \cdots, x_m, y), x_2, \cdots, x_m)^T - x$, we then have $g(x+t',y) = 0$, i.e. $t' \in Y(x,y)$. ☐

Lemma E.1.2. *If $T_a(\bar{z}) \neq \emptyset$ and TC holds at $z = \bar{z}$ for any $\bar{t} \in T_a(\bar{z})$, then there exists a neighborhood $V_{\bar{z}}$ such that $\forall z \in V_{\bar{z}}$ problem $\mathcal{I}^t(x,y)$ attains its minimum.*

Proof. From Lemma E.1.1, select a $t' \in Y(x,y) \neq \emptyset$. Define set

$$C = \{t \in \mathbb{R}^m \,|\, |t| \leq |t'|\}.$$

https://doi.org/10.51202/9783186952035 - Generiert durch IP 216.73.216.36, am 18.01.2026, 21:11:22. © Urheberrechtlich geschützter Inhalt. Ohne gesonderte
Erlaubnis ist jede urheberrechtliche Nutzung untersagt, insbesondere die Nutzung des Inhalts im Zusammenhang mit, für oder in KI-Systemen, KI-Modellen oder Generativen Sprachmodellen.

C is obviously a compact set. It is elementary to prove that the global minima of problem $\mathcal{I}^t(x, y)$ are identical as the global minima of

$$\min_{t \in Y(x,y) \cap C} t^T t. \tag{E.3}$$

Moreover, because $Y(x, y)$ is a closed set[1] and C is a compact set, problem (E.3) attains its global minima from the extreme value theorem. □

Lemma E.1.3. *Assume that $T_a(\bar{z}) \neq \emptyset$ and the TC holds at $z = \bar{z}$, we have*

$$\bar{z} \in F \Rightarrow \bar{z} \in F^m.$$

Proof. Assume that $\bar{z} \notin F^m$, i.e. $\exists\, t' \in \mathbb{R}^m$, such that

$$g(\bar{x} + t', \bar{y}) = 0, \tag{E.4}$$

$$t'^T t' < 1. \tag{E.5}$$

Eq. (E.4) indicates that $t' \in T_a(\bar{z})$. Eq. (E.5) is therefore a contradiction to the conclusion of Lemma 5.3.5. □

Note that under the conditions in the above lemma, $\bar{z} \in F^m \nRightarrow \bar{z} \in F$. An exemplary function is $g_0(\cdot, \cdot)$, satisfying

$$\begin{cases} g_0(\bar{x} + t, \bar{y}) < 0, \text{ if } t^T t < 1 \\ g_0(\bar{x} + t, \bar{y}) = 0, \text{ if } t^T t = 1 \\ g_0(\bar{x} + t, \bar{y}) > 0, \text{ if } t^T t > 1 \end{cases}.$$

Lemma E.1.4. *Assume that $\bar{z} \in F$, $T_a(\bar{z}) \neq \emptyset$ and the TC holds at $z = \bar{z}$. Then there exists a neighborhood $V_{\bar{z}}$, such that: (i) $V_{\bar{z}} \cap \bar{F} \neq \emptyset$, (ii) $V_{\bar{z}} \cap \bar{F}^m \neq \emptyset$.*

Proof. (i) Assume that $\bar{t} \in T_a(\bar{z})$. Denote $\delta = -r(\bar{x} + \bar{t}, \bar{y})$ as a unit direction (δ is well defined because the normal vector r is well defined under TC.). Denote $\bar{z}' = (\bar{x}'^T, \bar{y}'^T)^T$ with $\bar{x}' = \bar{x} + \epsilon\, \delta$ and $\bar{y}' = \bar{y}$. $\epsilon > 0 \in \mathbb{R}$. It is obvious that $\bar{z}' \to \bar{z}$, as $\epsilon \to 0$. So we need to prove that for $\epsilon > 0$ sufficiently small, $\bar{z}' \notin F$. Because

$$\begin{aligned} g(\bar{x}' + \bar{t}, \bar{y}') &= g(\bar{x} + \bar{t} + \epsilon\, \delta, \bar{y}) \\ &= g(\bar{x} + \bar{t}, \bar{y}) - \epsilon ||\nabla_x g(\bar{x} + \bar{t}, \bar{y})|| + o(\epsilon), \end{aligned}$$

and $||\nabla_x g(\bar{x} + \bar{t}, \bar{y})|| > 0$ (cf. TC), $g(\bar{x}' + \bar{t}, \bar{y}') < g(\bar{x} + \bar{t}, \bar{y}) = 0$ for sufficiently small ϵ. Moreover, because $\bar{t} \in T$, this leads to $\bar{z}' \notin F$.

(ii) Assume that $\bar{t} \in T_a(\bar{z})$. Denote $x' = \bar{x} + \epsilon \bar{t}$, $y' = \bar{x}$, $\epsilon > 0$. We prove that $z' = (x'^T, y'^T)^T \in \bar{F}^m$ for sufficiently small ϵ. Denote $t^* = \bar{t} - \epsilon \bar{t}$, we have $g(x' + t^*, y') = g(\bar{x} + \bar{t}, \bar{y}) = 0$, i.e. t^* is feasible to $\mathcal{I}^t(x', y')$. Moreover, $t^{*T} t^* = (1 - \epsilon)^2 \bar{t}^T \bar{t} = (1 - \epsilon)^2 < 1$ ($\bar{t}^T \bar{t} = 1$ from Lemma 5.3.5). Therefore $z' \notin F^m$, for ϵ sufficiency small. □

---

[1] This can be proved straightforwardly, since a closed set can be defined as a set which contains all its limit points.

https://doi.org/10.51202/9783186952035 - Generiert durch IP 216.73.216.36, am 18.01.2026, 21:11:22. © Urheberrechtlich geschützter Inhalt. Ohne gesonderte Erlaubnis ist jede urheberrechtliche Nutzung untersagt, insbesondere die Nutzung des Inhalts im Zusammenhang mit, für oder in KI-Systemen, KI-Modellen oder Generativen Sprachmodellen.

Lemma E.1.5. *Assume that $\bar{z} \in F$, $T_a(\bar{z}) \neq \emptyset$ and TC holds at $z = \bar{z}$. Select $z' = (x'^T, y'^T)^T \in V_{\bar{z}} \cap \bar{F}^m$ (such z' can always be found due to Lemma E.1.4) and denote t' as any global minimum of problem $\mathcal{I}^t(x', y')$ defined in Eq. (E.1) (t' can be attained due to Lemma E.1.2.). Then for $V_{\bar{z}}$ sufficiently small, we have*

$$\nabla_x g(x' + t', y') \neq 0. \tag{E.6}$$

Proof. Denote $V_{\bar{z}}^k = \{z \in \mathbb{R}^{m+n} \mid ||z - \bar{z}|| \leq 1/k\}$, $k = 1, \cdots, \infty$, as a sequence of neighborhoods. Select $z'_k \in V_{\bar{z}}^k \cap \bar{F}_m$ and denote t'_k as any global minimum of $\mathcal{I}^t(x'_k, y'_k)$. It is obvious that

$$x'_k \to \bar{x}, \tag{E.7a}$$
$$y'_k \to \bar{y}, \tag{E.7b}$$
$$g(x'_k + t'_k, y'_k) = 0. \tag{E.7c}$$

Assume that Eq. (E.6) does not hold, i.e., for k sufficiently large, we can always find x'_k and global minimum t'_k satisfying

$$\nabla_x g(x'_k + t'_k, y'_k) = 0. \tag{E.8}$$

Because $z'_k \in \bar{F}_m$, we have $t'^T_k t'_k < 1$. Therefore, t'_k, $k = 1, \cdots, \infty$, is a bounded sequence. Hence, there exists a sub-sequence $\Omega \subseteq \{1, \cdots, \infty\}$ such that t'_k, $k \in \Omega$ and $k \to \infty$, is convergent (cf. Bolzano-Weierstrass theorem). Denote $\bar{t}'$ as the limit value of this subsequence, i.e., $t'_k \to \bar{t}'$, for $k \in \Omega$ and $k \to \infty$.

From the continuity of $g(\cdot, \cdot)$ and Eq. (E.7c), we have

$$g(\bar{x} + \bar{t}', \bar{y}) = 0. \tag{E.9}$$

And from the continuity of $\nabla_x g(\cdot, \cdot)$ and Eq. (E.8), we have

$$\nabla_x g(\bar{x} + \bar{t}', \bar{y}) = 0. \tag{E.10}$$

Eq. (E.9) and Eq. (E.10) is a contradiction to the TC. □

Note that Eq. (E.6) indicates that LICQ condition of $\mathcal{I}^t(z)$ holds for z sufficiently closed to $\bar{z}$. Under the second-order sufficient conditions, this result can be actually obtained for more general parametric optimization problems (cf. Theorem D.3 in Appendix D). However, Lemma E.1.5 does not use the second-order sufficient conditions.

The following theorem builds up a link between the feasible set F of the the original SIP $\mathcal{P}$ with set F^m.

Theorem E.1.6. *Assume that $\bar{z} \in F$, $T_a(\bar{z}) \neq \emptyset$ and the TC holds at $z = \bar{z}$. Then for $V_{\bar{z}}$ sufficiently small, we have*

$$V_{\bar{z}} \cap F = V_{\bar{z}} \cap F^m.$$

Proof. (i) We first prove $V_{\bar{z}} \cap F^m \subseteq V_{\bar{z}} \cap F$, or equivalently $\overline{V_{\bar{z}} \cap F} \subseteq \overline{V_{\bar{z}} \cap F^m}$. From set operations

$$\overline{V_{\bar{z}} \cap F} = \bar{V}_{\bar{z}} \cup \bar{F} = \bar{V}_{\bar{z}} \cup (\bar{F}/\bar{V}_{\bar{z}}) = \bar{V}_{\bar{z}} \cup (\bar{F} \cap V_{\bar{z}}), \tag{E.11}$$
$$\overline{V_{\bar{z}} \cap F^m} = \bar{V}_{\bar{z}} \cup \bar{F}^m = \bar{V}_{\bar{z}} \cup (\bar{F}^m/\bar{V}_{\bar{z}}) = \bar{V}_{\bar{z}} \cup (\bar{F}^m \cap V_{\bar{z}}), \tag{E.12}$$

https://doi.org/10.51202/9783186952035 - Generiert durch IP 216.73.216.36, am 18.01.2026, 21:11:22. © Urheberrechtlich geschützter Inhalt. Ohne gesonderte Erlaubnis ist jede urheberrechtliche Nutzung untersagt, insbesondere die Nutzung des Inhalts im Zusammenhang mit, für oder in KI-Systemen, KI-Modellen oder Generativen Sprachmodellen.

so we only need to prove $z \in \bar{F} \cap V_{\bar{z}} \Rightarrow z \in \bar{F}^m \cap V_{\bar{z}}$ for $V_{\bar{z}}$ sufficiently small.

From Lemma 5.3.5, we have $0 \notin T_a(\bar{z})$ and therefore $g(\bar{x}, \bar{y}) = g(\bar{x} + 0, \bar{y}) > 0$. $\forall\, z' = (x'^T, y'^T)^T \in \bar{F} \cap V_{\bar{z}}$ (From Lemma E.1.4, $\bar{F} \cap V_{\bar{z}} \neq \emptyset$ and therefore we can always select z' for any sufficiently small $V_{\bar{z}}$.), because of the continuity of $g(\cdot,\cdot)$, for sufficiently small $V_{\bar{z}}$ we have

$$g(x', y') > 0. \tag{E.13}$$

Moreover, because $z' \in \bar{F}$ (i.e., z' is infeasible to $\mathcal{P}$), there exists a $t' \in T$ satisfying

$$g(x' + t', y') < 0, \tag{E.14a}$$

$$t'^T t' \leq 1. \tag{E.14b}$$

From Eqs. (E.13), (E.14a) and applying the mean value theorem, there exists a $\gamma \in (0, 1)$ such that

$$g(x' + \gamma t', y') = 0. \tag{E.15}$$

Because $\gamma^2 t'^T t' < t'^T t' \leq 1$, from the definition of F^m and Eq. (E.15), $z' = (x'^T, y'^T)^T \notin F^m$. Moreover, because z' is selected to to be in $V_{\bar{z}}$, we have $z' \in \bar{F}^m \cap V_{\bar{z}}$.

(ii) We then prove $V_{\bar{z}} \cap F \subseteq V_{\bar{z}} \cap F^m$, or equivalently $\overline{V_{\bar{z}} \cap F^m} \subseteq \overline{V_{\bar{z}} \cap F}$. From Eq. (E.11), we only need to prove $z \in \bar{F}^m \cap V_{\bar{z}} \Rightarrow z \in \bar{F} \cap V_{\bar{z}}$, for $V_{\bar{z}}$ sufficiently small.

From Lemma E.1.4, for sufficiently small $V_{\bar{z}}$, $\bar{F}^m \cap V_{\bar{z}} \neq \emptyset$ and therefore one can always select $z' = (x'^T, y'^T)^T \in \bar{F}^m \cap V_{\bar{z}}$. $\forall z' \in \bar{F}^m \cap V_{\bar{z}}$, denote t' as any global minimum of problem $\mathcal{I}^t(x', y')$ (t' can be attained due to Lemma E.1.2). Because $z' \in \bar{F}^m$,

$$t'^T t' < 1. \tag{E.16}$$

From Lemma E.1.5, Eq. (E.6) holds. Denote

$$\begin{aligned} \delta &= \frac{\nabla_x^T g(x' + t', y')}{||\nabla_x g(x' + t', y')||}, \\ t'' &= t' - \epsilon \delta. \end{aligned}$$

We have

$$\begin{aligned} g(x' + t'', y') &= g(x' + t' - \epsilon\delta, y') \\ &= g(x' + t', y') - \epsilon \nabla_x g(x' + t', y')\delta + o(\epsilon) \\ &= -\epsilon ||\nabla_x g(x' + t', y')|| + o(\epsilon) \\ &< 0, \end{aligned} \tag{E.17}$$

for $\epsilon > 0$ sufficiently small.

Moreover, because of Eq. (E.16), for ϵ sufficiently small, $||t''|| < 1$, i.e. t'' is feasible to $\mathcal{P}$. Therefore, from Eq. (E.17), $z' \notin F$. Because z' is selected to to be in $V_{\bar{z}}$, we have $z' \in \bar{F} \cap V_{\bar{z}}$. $\square$

E.2 Local equivalence of F^m and F^n

Having established the local equivalence between F and F^m through Theorem E.1.6. We now establish the local equivalence between F^m and F^n. Combining these two results will lead to Theorem 5.3.10 straightforwardly. Define

$$G^t(z) = \{t \in \mathbb{R}^m \,|\, t \text{ is a global minimum of } \mathcal{I}^t(z)\}.$$

https://doi.org/10.51202/9783186952035 - Generiert durch IP 216.73.216.36, am 18.01.2026, 21:11:22. © Urheberrechtlich geschützter Inhalt. Ohne gesonderte Erlaubnis ist jede urheberrechtliche Nutzung untersagt, insbesondere die Nutzung des Inhalts im Zusammenhang mit, für oder in KI-Systemen, KI-Modellen oder Generativen Sprachmodellen.

Lemma E.2.1. *Assume that* $\bar{z} \in F$, $T_a(\bar{z}) \neq \emptyset$ *and the TC holds at* $z = \bar{z}$, *we have*

$$G^t(\bar{z}) = T_a(\bar{z}).$$

Proof. (i) We first prove that $T_a(\bar{z}) \subseteq G^t(\bar{z})$. $\forall \bar{t} \in T_a(\bar{z})$, $g(\bar{x} + \bar{t}, \bar{y}) = 0$ (cf. the definition of $T_a(\bar{z})$) and $||\bar{t}|| = 1$ (cf. Lemma 5.3.5). Assume that $\bar{t} \notin G^t(\bar{z})$. Then there exists a $\bar{t}'$ such that $g(\bar{x} + \bar{t}', \bar{y}) = 0$ and $||\bar{t}'|| < ||\bar{t}|| = 1$. This is contradictory to the conclusion of Lemma 5.3.5.

(ii) We then prove that $G^t(\bar{z}) \subseteq T_a(\bar{z})$. If $\bar{t}'$ is a global minimizer of $\mathcal{I}^t(\bar{z})$, i.e., $\bar{t}' \in G^t(\bar{z})$, then $\bar{t}'$ is feasible to $\mathcal{I}^t(\bar{x}, \bar{y})$, i.e.

$$g(\bar{x} + \bar{t}', \bar{y}) = 0. \tag{E.18}$$

Moreover, because any $\bar{t} \in T_a(\bar{z})$ is feasible to problem $\mathcal{I}^t(\bar{z})$,

$$||\bar{t}'|| \leq ||\bar{t}|| = 1, \text{ i.e., } \bar{t}' \in T. \tag{E.19}$$

So from Eqs. (E.18), (E.19), $\bar{t}' \in T_a(\bar{z})$. □

Denote the KKT system of NLP $\mathcal{I}^t(z)$ as (cf. Theorem D.1 in Appendix D)

$$h^t(x, y, t, \beta) = \begin{pmatrix} 2t^T + \beta \nabla_x g(x + t, y) \\ g(x + t, y) \end{pmatrix} = 0, \tag{E.20}$$

where $\beta \in \mathbb{R}$ denotes the Lagrange multiplier for $\mathcal{I}^t(z)$. Note that we have used the property that $\nabla_t g(x + t, y) = \nabla_x g(x + t, y)$.

Lemma E.2.2. *Assume that* $\bar{z} \in F$, $T_a(\bar{z}) \neq \emptyset$ *and the TC holds at* $z = \bar{z}$. *We have: (i) LICQ and SC condition of* $\mathcal{I}^t(\bar{z})$ *hold,* $\forall \bar{t} \in T_a(\bar{z})$, *i.e., there exists a unique Lagrange multiplier* $\bar{\beta} = 2/||\nabla_x g(\bar{x} + \bar{t}, \bar{y})|| < 0$ *such that* $h(\bar{x}, \bar{y}, \bar{t}, \bar{\beta}) = 0$, $\forall \bar{t} \in T_a(\bar{z})$. *(ii) If we assume the SOSC (5.59) of* $\mathcal{I}(\bar{z})$ *holds in addition,* $\forall \bar{t} \in T_a(\bar{z})$, *then the SOSC of* $\mathcal{I}^t(\bar{z})$ *holds,* $\forall \bar{t} \in T_a(\bar{z})$.

Proof. (i) From Lemma E.2.1, any $\bar{t} \in T_a(\bar{z})$ is a local minimizer of $\mathcal{I}^t(\bar{z})$. LICQ condition of $\mathcal{I}^t(\bar{z})$ holds due to TC. From Theorem D.1 in Appendix D, there exists a unique Lagrange multiplier $\bar{\beta}$ satisfying Eq. (E.20). Because $||\bar{t}|| = 1$ (cf. Lemma 5.3.5), from the first equation in Eq. (E.20) we have

$$|\bar{\beta}| = \frac{2}{||\nabla_x g(\bar{x} + \bar{t}, \bar{y})||}.$$

The rest task is to prove that $\bar{\beta} > 0$. Because

$$\begin{aligned} g(\bar{x} + \bar{t} - \epsilon \bar{t}, \bar{y}) &= g(\bar{x} + \bar{t}, \bar{y}) - \epsilon \nabla_x g(\bar{x} + \bar{t}, \bar{y}) \bar{t} + o(\epsilon) \\ &= \frac{2\epsilon}{\bar{\beta}} \bar{t}^T \bar{t} + o(\epsilon) \\ &= \frac{2\epsilon}{\bar{\beta}} + o(\epsilon), \end{aligned}$$

if we assume that $\bar{\beta} < 0$, for $\epsilon > 0$ sufficiently small, $g(\bar{x} + (1 - \epsilon)\bar{t}, \bar{y}) < g(\bar{x} + \bar{t}, \bar{y}) = 0$. This is contradictory to $\bar{z} \in F$.

https://doi.org/10.51202/9783186952035 - Generiert durch IP 216.73.216.36, am 18.01.2026, 21:11:22. © Urheberrechtlich geschützter Inhalt. Ohne gesonderte
Erlaubnis ist jede urheberrechtliche Nutzung untersagt, insbesondere die Nutzung des Inhalts im Zusammenhang mit, für oder in KI-Systemen, KI-Modellen oder Generativen Sprachmodellen.

(ii) Since SC condition holds, we need to prove that (cf. Eq. (D.7) in Appendix D) $s^T \nabla_{tt} L^t(\bar{x}, \bar{y}, \bar{t}, \bar{\beta}) s > 0$, for all $s \neq 0$ such that $\nabla_t g(\bar{x} + \bar{t}, \bar{y}) s = 0$, where $L^t(x, y, t, \beta) = t^T t + \beta g(x + t, y)$ is the Lagrange function of $\mathcal{I}^t(z)$. From Eqs. (5.55), (5.54),

$$\{s \in \mathbb{R}^m \,|\, \bar{t}^T s = 0, s \neq 0\} = \{s \in \mathbb{R}^m \,|\, \nabla_t g(\bar{x} + \bar{t}, \bar{y}) s = 0, s \neq 0\}.$$

Moreover,

$$\nabla_{tt} L^t(\bar{x}, \bar{y}, \bar{t}, \bar{\beta}) = 2I + \bar{\beta} \nabla_{tt} g(\bar{x} + \bar{t}, \bar{y}) = 2I + \frac{\nabla_{tt} g(\bar{x} + \bar{t}, \bar{y})}{\bar{l}} = \frac{W(\bar{x}, \bar{y}, \bar{t}, \bar{l})}{\bar{l}},$$

where $\bar{l}$ is defined in Eq. (5.54) and $W(\cdot)$ is defined in Eq. (5.58). The SOSC of $\mathcal{I}^t(\bar{z})$ follows directly from the SOSC of $\mathcal{I}(\bar{z})$. □

If the assumptions in the previous lemma hold, we can denote $T_a(\bar{z}) = \{\bar{t}^1, \cdots, \bar{t}^J\}$ as a finite set (cf. Lemma 5.3.7), where $\bar{t}^j$ are global minima of $\mathcal{I}^t(\bar{z})$ (cf. Lemma E.2.1). Denote $\bar{\beta}^j$, $j = 1, \cdots, J$, as the associated Lagrange multipliers of $\bar{t}^j$. By applying Theorem D.3 in Appendix D to $\mathcal{I}^t(x, y)$, Eq. (E.20) therefore locally determines $\mathcal{C}$-functions $\tilde{t}^j(x, y)$, $\beta^j(x, y)$, satisfying $\tilde{t}^j(\bar{x}, \bar{y}) = \bar{t}^j$, $\beta^j(\bar{x}, \bar{y}) = \bar{\beta}^j$, $\forall\, j = 1, \cdots, J$. Moreover, $\tilde{t}^j(x, y)$ are unique local minima of $\mathcal{I}^t(x, y)$ with associated unique Lagrange multipliers $\beta^j(x, y)$.

Corollary E.2.3. *From the assumptions in Lemma E.2.2, there exists a neighborhood $V_{\bar{z}}$ such that*

$$V_{\bar{z}} \cap F^m = V_{\bar{z}} \cap \{z = (x^T, y^T)^T \,|\, \tilde{t}^{jT}(x, y) \tilde{t}^j(x, y) \geq 1, j = 1, \cdots, J\}, \tag{E.21}$$

where $\tilde{t}^j(\cdot, \cdot)$, $j = 1, \cdots, J$, are implicitly defined by Eq. (E.20).

Proof. Directly from the local reduction theorem 3.3.3 in [67]. □

Recall that $t^j(x, y)$, $d^j(x, y)$, $j = 1, \cdots, J$, are implicitly defined functions of $h(x, y, t, d) = 0$ in Eq. (5.56), we have the following lemma.

Lemma E.2.4. *Assume $\bar{z} \in F$, $T_a(\bar{z}) \neq \emptyset$, TC holds at $z = \bar{z}$ and the SOSC (5.59) of $\mathcal{I}(\bar{z})$ is fulfilled for all $\bar{t} \in T_a(\bar{z})$. Then there exists a neighborhood $V_{\bar{z}}$ such that $\forall (x^T, y^T)^T \in V_{\bar{z}}$,*

$$\begin{aligned} \tilde{t}^j(x, y) &= t^j(x, y),\ j = 1, \cdots, J, \\ \beta^j(x, y) &= \frac{2 d^j(x, y)}{||\nabla_x g(x + t^j(x, y), y)||},\ j = 1, \cdots, J, \end{aligned} \tag{E.22}$$

or equivalently,

$$\begin{aligned} t^j(x, y) &= \tilde{t}^j(x, y),\ j = 1, \cdots, J, \\ d^j(x, y) &= \frac{1}{2} \beta^j(x, y) ||\nabla_x g(x + \tilde{t}^j(x, y), y)||,\ j = 1, \cdots, J. \end{aligned} \tag{E.23}$$

Proof. We prove only Eq. (E.22). Eq. (E.23) follows straightforwardly from Eq. (E.22). Denote

$$\begin{aligned} &\mathcal{X}^j(x, y) = (\tilde{t}^{jT}(x, y), \beta^j(x, y))^T,\ j = 1, \cdots, J, \\ &\mathcal{Y}^j(x, y) = (t^{jT}(x, y), 2 d^j(x, y) / ||\nabla_x g(x + t^j(x, y), y)||)^T,\ j = 1, \cdots, J. \end{aligned}$$

https://doi.org/10.51202/9783186952035 - Generiert durch IP 216.73.216.36, am 18.01.2026, 21:11:22. © Urheberrechtlich geschützter Inhalt. Ohne gesonderte Erlaubnis ist jede urheberrechtliche Nutzung untersagt, insbesondere die Nutzung des Inhalts im Zusammenhang mit, für oder in KI-Systemen, KI-Modellen oder Generativen Sprachmodellen.

From the definition of $\tilde{t}^j(\cdot)$ and $\beta^j(\cdot)$,

$$h^t(x, y, \mathcal{X}^j(x, y)) = 0.$$

Assume now that Eq. (E.22) does not hold, i.e. $\mathcal{X}^j(\cdot, \cdot) \neq \mathcal{Y}^j(\cdot, \cdot)$. Because $t^j(x, y)$, $d^j(x, y)$ satisfy Eq. (5.56), one can straightforwardly check that

$$\begin{aligned} &\mathcal{X}^j(\bar{x}, \bar{y}) = \mathcal{Y}^j(\bar{x}, \bar{y}), \\ &h^t(x, y, \mathcal{Y}^j(x, y)) = 0. \end{aligned}$$

That is, both functions $\mathcal{X}^j(x, y)$ and $\mathcal{Y}^j(x, y)$ fulfill $h^t(x, y, \cdot, \cdot) = 0$ and they have the same value at $z = \bar{z}$. According to the IFT, however, Eq. (E.20) locally *uniquely* determines an implicitly function near $z = \bar{z}$, which is a contradiction. ☐

Theorem E.2.5. *Assume $\bar{z} \in F$, $T_a(\bar{z}) \neq \emptyset$, TC holds at $z = \bar{z}$ and the SOSC (5.59) of $\mathcal{I}(\bar{x}, \bar{y})$ is fulfilled for all $\bar{t} \in T_a(\bar{z})$, there exists a neighborhood $V_{\bar{z}}$ such that*

$$V_{\bar{z}} \cap F^m = V_{\bar{z}} \cap F^n. \tag{E.24}$$

Proof. We prove that there exists a neighborhood $V_{\bar{z}}$ such that

$$d^j(x, y) = ||\tilde{t}^j(x, y)||, \ j = 1, \cdots, J. \tag{E.25}$$

From Eq. (E.20), we have

$$|\beta^j(x, y)| = \frac{2||\tilde{t}^j(x, y)||}{||\nabla_x g(x + \tilde{t}^j(x, y), y)||}.$$

Because $\beta^j(\bar{x}, \bar{y}) = 2/||\nabla_x g(\bar{x} + \bar{t}^j, \bar{y})|| > 0$ (cf. Lemma E.2.2), from the IFT the implicitly defined funtion

$$\beta^j(x, y) = \frac{2||\tilde{t}^j(x, y)||}{||\nabla_x g(x + \tilde{t}^j(x, y), y)||}.$$

Substitute this equation into Eq. (E.23) will lead to Eq. (E.25). ☐

Theorem 5.3.10 follows as a direct consequence of Theorem E.2.5 and Theorem E.1.6.

https://doi.org/10.51202/9783186952035 - Generiert durch IP 216.73.216.36, am 18.01.2026, 21:11:22. © Urheberrechtlich geschützter Inhalt. Ohne gesonderte Erlaubnis ist jede urheberrechtliche Nutzung untersagt, insbesondere die Nutzung des Inhalts im Zusammenhang mit, für oder in KI-Systemen, KI-Modellen oder Generativen Sprachmodellen.

Bibliography

[1] L. Achenie and L. Biegler. Algorithmic synthesis of chemical reactor networks using mathematical-programming. *Industrial & Engineering Chemistry Fundamentals*, 25: 621–627, 1986.

[2] L. Achenie and L. Biegler. A superstructure based approach to chemical reactor network synthesis. *Computers & Chemical Engineering*, 14:23–40, 1990.

[3] F. Al-Khayyal and J. Falk. Jointly constrained biconvex programming. *Mathematics of Operations Research*, 8(2):273–286, 1983.

[4] A. Andrew, K. Chu, and P. Lancaster. Derivatives of eigenvalues and eigenvectors of matrix functions. *SIAM Journal on Matrix Analysis and Applications*, 14(4): 903–926, 1993.

[5] M. Anitescu. On using the elastic mode in nonlinear programming approaches to mathematical programs with complementarity constraints. *SIAM Journal on Optimization*, 15(4):1203–1236, 2005.

[6] M. Anitescu, P. Tseng, and S. Wright. Elastic-mode algorithms for mathematical programs with equilibrium constraints: global convergence and stationarity properties. *Mathematical Programming*, 110:337–371, 2005.

[7] F. Assassa and W. Marquardt. Dynamic optimization using adaptive direct multiple shooting. *Computers & Chemical Engineering*, 60:242 – 259, 2014.

[8] A. Bagirov, N. Karmitsa, and M. Mäkelä. *Introduction to Nmonsmooth Optimization: Theory, Practice and Software.* Springer Publishing Company, 2014.

[9] P. Bahri, J. Bandoni, and J. Romagnoli. Effects of disturbances in optimizing control: steady-state open-loop backoff problem. *AIChE Journal*, 42:983–994, 1996.

[10] L. Bakule. Decentralized control: an overview. *Annual Reviews in Control*, 32(1):87 – 98, 2008.

[11] S. Balakrishna and L. Biegler. Constructive targeting approaches for the synthesis of chemical ractor networks. *Industrial & Engineering Chemistry Research*, 31(1): 300–312, 1992.

[12] S. Balakrishna and L. Biegler. An unified approach for the simultaneous synthesis of reaction, energy, and separation systems. *Industrial & Engineering Chemistry Research*, 32(7):1372–1382, 1993.

[13] E. Balas. Disjunctive programming and a hierarchy of relaxations for discrete optimization problems. *SIAM Journal on Algebraic and Discrete Methods*, 6(3):466–486, 1985.

https://doi.org/10.51202/9783186952035 - Generiert durch IP 216.73.216.36, am 18.01.2026, 21:11:22. © Urheberrechtlich geschützter Inhalt. Ohne gesonderte Erlaubnis ist jede urheberrechtliche Nutzung untersagt, insbesondere die Nutzung des Inhalts im Zusammenhang mit, für oder in KI-Systemen, KI-Modellen oder Generativen Sprachmodellen.

[14] B. Baumrucker and L. Biegler. MPEC strategies for optimization of a class of hybrid dynamic systems. *Journal of Process Control*, 19(8):1248–1256, 2009.

[15] B. Baumrucker, J. Renfro, and L. Biegler. MPEC problem formulations and solution strategies with chemical engineering applications. *Computers & Chemical Engineering*, 32(12):2903 – 2913, 2008.

[16] B. Baumrucker, J. Renfro, and L. Biegler. MPEC problem formulations in chemical engineering applications. *Computers & Chemical Engineering*, 32:2903–2913, 2008.

[17] P. Belotti, C. Kirches, S. Leyffer, J. Linderoth, J. Luedtke, and A. Mahajan. Mixed-integer nonlinear optimization. *Acta Numerica*, 22:1–131, 5 2013.

[18] B. Bhattacharjee, P. Lemonidis, W. Green Jr., and P. Barton. Global solution of semi-infinite programs. *Mathematical Programming*, 103(2):283–307, 2005.

[19] L. Biegler, I. Grossmann, and A. Westerberg. *Systematic methods of chemical process design*. Prentice Hall, 1997.

[20] T. Binder, L. Blank, H. Bock, R. Bulirsch, W. Dahmen, M. Diehl, T. Kronseder, W. Marquardt, J. Schlöder, and O. von Stryk. Introduction to model based optimization of chemical processes on moving horizons. In M. Grötschel, S. Krumke, and J Rambau, editors, *Online Optimization of Large Scale Systems*, pages 295–339. Springer Berlin Heidelberg, Berlin, Heidelberg, 2001.

[21] A. Blanco and J. Bandoni. Interaction between process design and process operability of chemical processes: an eigenvalue optimization approach. *Computers & Chemical Engineering*, 27:1291–1301, 2003.

[22] P. Bonami, M. Kilinc, and J. Linderoth. Algorithms and software for convex mixed integer nonlinear programs. In J. Lee and S. Leyffer, editors, *Mixed Integer Nonlinear Programming*, volume 154 of *The IMA Volumes in Mathematics and its Applications*, pages 1–39. Springer New York, 2012.

[23] B. Borches and J. Mitchell. An improved branch-and-bound algorithm for mixed-integer nonlinear programs. *Computers & Operations Research*, 21(4):359–367, 1994.

[24] E. Bristol. On a new measure of interaction for multivariable process control. *IEEE transactions on automatic control*, AC-11(1):133–134, 1966.

[25] J. Burke, A. Lewis, and M. Overton. Two numerical methods for optimizing matrix stability. *Linear Algebra and its Applications*, 351-352:117–145, 2002.

[26] J. Burke, A. Lewis., and M. Overton. Optimization and pseudospectra, with applications to robust stability. *SIAM Journal on Matrix Analysis and Applications*, 25 (1):80–104, 2003.

[27] J. Burke, A. Lewis, and M. Overton. A robust gradient sampling algorithm for nonsmooth, nonconvex optimization. *SIAM Journal on Optimization*, 15(3):751–779, 2005.

https://doi.org/10.51202/9783186952035 - Generiert durch IP 216.73.216.36, am 18.01.2026, 21:11:22. © Urheberrechtlich geschützter Inhalt. Ohne gesonderte Erlaubnis ist jede urheberrechtliche Nutzung untersagt, insbesondere die Nutzung des Inhalts im Zusammenhang mit, für oder in KI-Systemen, KI-Modellen oder Generativen Sprachmodellen.

[28] Y. Cao and T. Marlin. Control structure design to achieve multiple performance criteria. In *Proceedings of the 7th International Symposium on DYCOPS, Boston*, 2004.

[29] M. Chiu and Y. Arkun. Decentralized control structure selection based on integrity considerations. *Industrial & Engineering Chemistry Research*, 29(3):369–373, 1990.

[30] I. Coope and G. Watson. A projected lagrangian algorithm for semi-infinite programming. *Mathematical Programming*, 32(3):337–356, 1985.

[31] R. Dakin. A tree-search algorithm for mixed integer programming problems. *The Computer Journal*, 8(3):250–255, 1965.

[32] A. Demiguel, M. Friedlander, F. Nogales, and S. Scholtes. A two-sided relaxation scheme for mathematical programs with equilibrium constraints. *SIAM Journal on Optimization*, 16(1):587–609, 2005.

[33] A. Dhooge, W. Govaerts, and Yu. A. Kuznetsov. MatCont: A MATLAB package for numerical bifurcation analysis of ODEs. *ACM TOMS*, 29:141–164, 2003.

[34] M. Diehl, J. Gerhard, W. Marquardt, and M. Mönnigmann. Numerical solution approaches for robust nonlinear optimal control problems. *Computers & Chemical Engineering*, 32(6):1279–1292, 2008.

[35] J. Douglas. *Conceptual Design of Chemical Processes*. McGraw-Hill, New York, 2nd edition, 1988.

[36] M. Duran and I. Grossmann. An outer-approximation algorithm for a class of mixed-integer nonlinear programs. *Mathematical Programming*, 36(3):307–339, 1986.

[37] F. Facchinei and J. Pang. *Finite-dimensional Variational Inequalities and Complementarity Problems*. Springer New York, 2003.

[38] F. Facchinei, H. Jiang, and L. Qi. A smoothing method for mathematical programs with equilibrium constraints. *Mathematical Programming*, 85(1):107–134, 1999.

[39] M. Feinberg and D. Hildebrandt. Optimal reactor design from a geometric viewpoint: universal properties of the attainable region. *Chemical Engineering Science*, 52(10): 1637–1665, 1997.

[40] A. Fiacco. Sensitivity analysis for nonlinear programming using penalty methods. *Mathematical Programming*, 10(1):287–311, 1976.

[41] A. Fiacco and Y. Ishizuka. Sensitivity and stability analysis for nonlinear programming. *Annals of Operations Research*, 27(1):215–235, 1990.

[42] R. Fletcher and S. Leyffer. Solving mixed integer nonlinear programs by outer approximation. *Mathematical Programming*, 66(1-3):327–349, 1994.

[43] R. Fletcher and S. Leyffer. Solving mathematical programs with complementarity constraints as nonlinear programs. *Optimization Methods and Software*, 19(1):15–40, 2004.

https://doi.org/10.51202/9783186952035 - Generiert durch IP 216.73.216.36, am 18.01.2026, 21:11:22. © Urheberrechtlich geschützter Inhalt. Ohne gesonderte Erlaubnis ist jede urheberrechtliche Nutzung untersagt, insbesondere die Nutzung des Inhalts im Zusammenhang mit, für oder in KI-Systemen, KI-Modellen oder Generativen Sprachmodellen.

[44] R. Fletcher, S. Leyffer, D. Ralph, and S. Scholtes. Local convergence of SQP methods for mathematical programs with equilibrium constraints. *SIAM Journal on Optimization*, 17(1):259–286, 2006.

[45] C. Floudas. *Nonlinear and Mixed-Integer Optimization: Fundamentals and Applications.* Oxford University Press, 1995.

[46] C. Floudas. *Deterministic Global Optimization: Theory, Methods and Applications.* Springer Verlag, Secaucus, USA, 2005.

[47] C. Floudas and C. Gounaris. A review of recent advances in global optimization. *Journal of Global Optimization*, 45(1):3–38, 2009.

[48] H. Fogler. *Elements of Chemical Reaction Engineering.* Prentice Hall, 1991.

[49] G. Franklin, D. Powell, and A. Emami-Naeini. *Feedback Control of Dynamic Systems.* Prentice Hall PTR, Upper Saddle River, NJ, USA, 4th edition, 2001.

[50] H. Freund and K. Sundmacher. Towards a methodology for the systematic analysis and design of efficient chemical processes: Part 1. from unit operations to elementary process functions. *Chemical Engineering and Processing: Process Intensification*, 47 (12):2051 – 2060, 2008.

[51] E. Gagnon, A. Desbiens, and A. Pomericau. Selection of pairing and constrained robust decentralized pi controllers. In *Proceedings of the 1999 American Control Conference*, pages 4343 – 4347. American Automatic Control Council, IFAC, 1999.

[52] A. Geoffrion. Generalized benders decomposition. *Journal of Optimization Theory and Applications*, 10(4):237–260, 1972.

[53] J. Gerhard, W. Marquardt, and M. Mönnigmann. Normal vectors on critical manifolds for robust design of transient processes in the presence of fast disturbances. *SIAM Journal on Applied Dynamical Systems*, 7(2):461–490, 2008.

[54] P. Gill, W. Murray, and M. Saunders. User's guide for SNOPT 5.3: a FORTRAN package for large-scale nonlinear programming. Technical report, Systems Optimization Laboratory, Stanford, USA, 1998.

[55] D. Glasser, C. Crowe, and D. Hildebrandt. A geometric approach to steady flow reactors: the attainable region and optimization in concentration space. *Industrial & Engineering Chemistry Research*, 26(9):1803–1810, 1987.

[56] V. Gopal and L. Biegler. Smoothing methods for complementarity problems in process engineering. *AIChE Journal*, 45(7):1535–1547, 1999.

[57] I. Grossmann. Review of nonlinear mixed-integer and disjunctive programming techniques. *Optimization and Engineering*, 3(3):227–252, 2002.

[58] O. Gupta and A. Ravindran. Branch and bound experiments in convex nonlinear integer programming. *Management Science*, 31(12):1533–1546, 1985.

https://doi.org/10.51202/9783186952035 - Generiert durch IP 216.73.216.36, am 18.01.2026, 21:11:22. © Urheberrechtlich geschützter Inhalt. Ohne gesonderte Erlaubnis ist jede urheberrechtliche Nutzung untersagt, insbesondere die Nutzung des Inhalts im Zusammenhang mit, für oder in KI-Systemen, KI-Modellen oder Generativen Sprachmodellen.

[59] S. Gustafson. A three-phase algorithm for semi-infinite programs. In A. Fiacco and K. Kortanek, editors, *Semi-Infinite Programming and Applications*, volume 215 of *Lecture Notes in Economics and Mathematical Systems*, pages 138–157. Springer Berlin Heidelberg, 1983.

[60] S. Han. Globally convergent method for nonlinear-programming. *Journal of Optimization Theory and Applications*, 22(3):297–309, 1977.

[61] R. Hannemann and W. Marquardt. Continuous and discrete composite adjoints for the Hessian of the Lagrangian in shooting algorithms for dynamic optimization. *SIAM Journal on Scientific Computing*, 31(6):4675–4695, 2010.

[62] J. Heath, I. Kookos, and J. Perkins. Process control structure selection based on economics. *AIChE Journal*, 46(10):1998–2016, 2000.

[63] M. Herty and S. Steffensen. MPCC solution approaches for a class of MINLPs with applications in chemical engineering. *Technical Reports of Aachen Institute for Advanced Study in Computational Engineering Science*, 2012.

[64] R. Hettich. An implementation of a discretization method for semi-infinite programming. *Mathematical Programming*, 34(3):354–361, 1986.

[65] R. Hettich and H. Jongen. Semi-infinite programming: Conditions of optimality and applications. In J. Stoer, editor, *Optimization Techniques*, volume 7 of *Lecture Notes in Control and Information Sciences*, pages 1–11. Springer Berlin Heidelberg, 1978.

[66] R. Hettich and K. Kortanek. Semi-infinite programming: theory, methods, and applications. *SIAM Review*, 35(3):380–429, 1993.

[67] R. Hettich and P. Zencke. *Numerische Methoden der Approximation und semi-infiniten Optimierung*. Teubner Studienbücher Mechanik. Vieweg+Teubner Verlag, 1982.

[68] D. Hildebrandt, D. Glasser, and C. Crowe. Geometry of the attainable region generated by reaction and mixing: with and without constraints. *Industrial & Engineering Chemistry Research*, 29(1):49–58, 1990.

[69] K. Holmstrom. TOMLAB - an environment for solving optimization problems in Matlab. Technical report, Proceedings for the Nordic Matlab Conference 97, October 27-28, Stockholm, 1997.

[70] F. Horn. Attainable and non-attainable regions in chemical reaction technique. In *3rd European Symposium on Chemical Reaction Engineering, Pergamon Press, New York*, page 1, 1965.

[71] M. Hovd and S. Skogestad. Pairing criteria for decentralized control of unstable plants. *Industrial & Engineering Chemistry Research*, 33:2134–2139, 1994.

[72] X. Hu and D. Ralph. Convergence of a penalty method for mathematical programming with complementarity constraints. *Journal of Optimization Theory and Applications*, 123(2):365–390, 2004.

https://doi.org/10.51202/9783186952035 - Generiert durch IP 216.73.216.36, am 18.01.2026, 21:11:22. © Urheberrechtlich geschützter Inhalt. Ohne gesonderte Erlaubnis ist jede urheberrechtliche Nutzung untersagt, insbesondere die Nutzung des Inhalts im Zusammenhang mit, für oder in KI-Systemen, KI-Modellen oder Generativen Sprachmodellen.

[73] R. Jackson. Optimization of chemical reactors with respect to flow configuration. *Journal of Optimization Theory and Applications*, 2(4):240–259, 1968.

[74] H. Jiang and D. Ralph. Smooth SQP methods for mathematical programs with nonlinear complementarity constraints. *SIAM Journal on Optimization*, 10(3):779–808, 2000.

[75] H. Jongen. Parametric optimization: critical points and local minima. *American Mathematical Society, Lectures in Applied Mathematics*, 26:317–335, 1990.

[76] H. Jongen. Theoretical background. In *Parametric Optimization: Singularities, Pathfollowing and Jumps*, pages 20–55. Vieweg+Teubner Verlag, 1990.

[77] H. Jongen and O. Stein. Smoothing by mollifiers (part I): semi-infinite optimization. *Journal of Global Optimization*, 41:319–334, 2008.

[78] H. Jongen and O. Stein. Smoothing by mollifiers (part II): nonlinear optimization. *Journal of Global Optimization*, 41:335–350, 2008.

[79] H. Jongen, P. Jonker, and F. Twilt. Critical sets in parametric optimization. *Mathematical Programming*, 34(3):333–353, 1986.

[80] H. Jongen, J. Rückmann, and O. Stein. Generalized semi-infinite optimization: a first order optimality condition and examples. *Mathematical Programming*, 83(1): 145–158, 1998.

[81] J. Jorgensen and S. Jorgensen. Towards automatic decentralized control structure selection. *Computers & Chemical Engineering*, 24:841–846, 2000.

[82] D. Kastsian and M. Mönnigmann. Robust optimization of fixed points of nonlinear discrete time systems with uncertain parameters. *SIAM Journal on Applied Dynamical Systems*, 9(2):357–390, 2010.

[83] A. Khaki-Sedigh and B. Moaveni. *Control Configuration Selection for Multivariable Plants*, volume 391. Springer Berlin Heidelberg, 2009.

[84] H. Khalil. *Nonlinear systems.* Pearson Education International Inc., New Jersey, 3rd edition, 2000.

[85] V. Klema and A. Laub. The singular value decomposition: its computation and some applications. *IEEE Transactions on Automatic Control*, 25(2):164–176, 1980.

[86] A. Kokossis and C. Floudas. Optimization of complex reactor networks 1: Isothermal operation. *Chemical Engineering Science*, 45(3):595–614, 1990.

[87] A. Kokossis and C. Floudas. Stability in optimal design: Synthesis of complex reactor networks. *AIChE Journal*, 40(5):849–861, 1994.

[88] A. Kokossis and C. Floudas. Optimization of conplex reactor networks 2: Non-isothermal operation. *Chemical Engineering Science*, 49(7):1037–1051, 1994.

https://doi.org/10.51202/9783186952035 - Generiert durch IP 216.73.216.36, am 18.01.2026, 21:11:22. © Urheberrechtlich geschützter Inhalt. Ohne gesonderte
Erlaubnis ist jede urheberrechtliche Nutzung untersagt, insbesondere die Nutzung des Inhalts im Zusammenhang mit, für oder in KI-Systemen, KI-Modellen oder Generativen Sprachmodellen.

[89] N. Konda, G. Rangaiah, and P. Krishnaswamy. Plantwide control of industrial processes: an integrated framework of simulation and heuristics. *Industrial & Engineering Chemistry Research*, 44(22):8300–8313, 2005.

[90] I. Kookos and J. Perkins. Heuristic-based mathematical programming framework for control structure selection. *Industrial & Engineering Chemistry Research*, 40(9): 2079–2088, 2001.

[91] D. Kressner and B. Vandereycken. Subspace methods for computing the pseudospectral abscissa and the stability radius. *SIAM Journal on Matrix Analysis and Applications*, 35(1):292–313, 2014.

[92] A. Lakshmanan and L. Biegler. Synthesis of optimal chemical reactor networks. *Industrial & Engineering Chemistry Research*, 35(4):1344–1353, 1996.

[93] J. Lee and M. Morari. Robust measurement selection. *Automatica*, 27(3):519–527, 1991.

[94] S. Lee and I. Grossmann. New algorithms for nonlinear generalized disjunctive programming. *Computers & Chemical Engineering*, 24(9-10):2125–2141, 2000.

[95] S. Lee and I. Grossmann. A global optimization algorithm for nonconvex generalized disjunctive programming and applications to process systems. *Computers & Chemical Engineering*, 25(11-12):1675–1697, 2001.

[96] P. Lemonidis. *Global optimization algorithms for semi-infinite and generalized semi-infinite programs.* PhD thesis, Massachusetts Institute of Technology, Dept. of Chemical Engineering, 2008.

[97] S. Leyffer. Integrating SQP and branch-and-bound for mixed integer nonlinear programming. *Computational Optimization and Applications*, 18(3):295–309, 2001.

[98] S. Leyffer. Complementarity constraints as nonlinear equations: Theory and numerical experience. In S. Dempe and V. Kalashnikov, editors, *Optimization with Multivalued Mappings*, volume 2 of *Springer Optimization and Its Applications*, pages 169–208. Springer US, 2006.

[99] S. Leyffer and A. Mahajan. Nonlinear constrained optimization: Methods and software. Technical report, Argonne National Laboratory, USA, 2010.

[100] L. Liberti and C. Pantelides. Convex envelopes of monomials of odd degree. *Journal of Global Optimization*, 25(2):157–168, 2003.

[101] G. Lin and M. Fukushima. New relaxation method for mathematical programs with complementarity constraints. *Journal of Optimization Theory and Applications*, 118 (1):81–116, 2003.

[102] G. Lin and M. Fukushima. A modified relaxation scheme for mathematical programs with complementarity constraints. *Annals of Operations Research*, 133(1-4):63–84, 2005.

https://doi.org/10.51202/9783186952035 - Generiert durch IP 216.73.216.36, am 18.01.2026, 21:11:22. © Urheberrechtlich geschützter Inhalt. Ohne gesonderte
Erlaubnis ist jede urheberrechtliche Nutzung untersagt, insbesondere die Nutzung des Inhalts im Zusammenhang mit, für oder in KI-Systemen, KI-Modellen oder Generativen Sprachmodellen.

[103] Y. Lin and M. Stadtherr. Deterministic global optimization of nonlinear dynamic systems. *AIChE Journal*, 53(4):866–875, 2007.

[104] X. Liu and J. Sun. Generalized stationary points and an interior-point method for mathematical programs with equilibrium constraints. *Mathematical Programming*, 101(1):231–261, 2004.

[105] M. Lopez and G. Still. Semi-infinite programming. *European Journal of Operational Research*, 180(2):491–518, 2007.

[106] J. Lunze. *Regelungstechnik 1: Systemtheoretische Grundlagen, Analyse und Entwurf einschleifiger Regelungen.* Springer Berlin Heidelberg, Berlin, Heidelberg, 2005.

[107] Z. Luo, J. Pang, and D. Ralph. *Mathematical programs with equilibrium constraints.* Cambridge University Press, 1996.

[108] Z. Luo, J. Pang, D. Ralph, and S. Wu. Exact penalization and stationarity conditions of mathematical programs with equilibrium constraints. *Mathematical Programming*, 75(1):19–76, 1996.

[109] M. Luyben, B. Tyreus, and W. Luyben. Plantwide control design procedure. *AIChE Journal*, 43(12):3161–3174, 1997.

[110] W. Luyben. Heuristic design of reaction/separation processes. *Industrial & Engineering Chemistry Research*, 49(22):11564–11571, 2010.

[111] W. Luyben, B. Tyreus, and M. Luyben. *Plantwide Process Control.* McGraw-Hill, 1998.

[112] M. Marden. *Geometry of Polynomials, 2nd edition.* American Mathematical Society, 1966.

[113] W. Marquardt and M. Mönnigmann. Constructive nonlinear dynamics in process systems engineering. *Computer & Chemical Engineering*, 29:1265–1275, 2005.

[114] T. Mcavoy. Synthesis of plantwide control systems using optimization. *Industrial & Engineering Chemistry Research*, 38(8):2984–2994, 1999.

[115] A. Mitsos and A. Tsoukalas. Global optimization of generalized semi-infinite programs via restriction of the right hand side. *Journal of Global Optimization*, pages 1–17, 2014.

[116] A. Mitsos and A. Tsoukalas. Global optimization of generalized semi-infinite programs via restriction of the right hand side. *Journal of Global Optimization*, 61(1): 1–17, 2015.

[117] M. Mohideen, J. Perkins, and E. Pistikopoulos. Optimal design of dynamic systems under uncertainty. *AIChE Journal*, 42(8):2251–2272, 1996.

[118] M. Mönnigmann. *Constructive Nonlinear Dynamics for the Design of Chemical Engineering Processes.* PhD thesis, RWTH Aachen University, 2004.

https://doi.org/10.51202/9783186952035 - Generiert durch IP 216.73.216.36, am 18.01.2026, 21:11:22. © Urheberrechtlich geschützter Inhalt. Ohne gesonderte Erlaubnis ist jede urheberrechtliche Nutzung untersagt, insbesondere die Nutzung des Inhalts im Zusammenhang mit, für oder in KI-Systemen, KI-Modellen oder Generativen Sprachmodellen.

[119] M. Mönnigmann and W. Marquardt. Normal vectors on manifolds of critical points for parametric robustness of equilibrium solutions of ODE systems. *Journal of Nonlinear Science*, 12(2):85–112, 2002.

[120] M. Mönnigmann and W. Marquardt. Steady state process optimization with guaranteed robust stability and robust feasibility. *AIChE Journal*, 49(12):3110–3126, 2003.

[121] D. Muñoz and W. Marquardt. A normal vector approach for integrated process and control design with uncertain model parameters and disturbances. *Computers and Chemical Engineering*, 2012.

[122] D. Muñoz and W. Marquardt. Robust design of closed-loop nonlinear systems with input and state constraints. *Annual Reviews in Control*, 37:232–245, 2013.

[123] D. Murthy and R. Haftka. Derivatives of eigenvalues and eigenvectors of a general complex matrix. *International Journal for Numerical Methods in Engineering*, 26 (2):293–311, 1988.

[124] L. Narraway and J. Perkins. Selection of process-control structure based on economics. *Computers & Chemical Engineering*, 18(Suppl. S):S511–S515, 1994.

[125] G. Nemhauser and L. Wolsey. *Integer and Combinatorial Optimization*. Wiley-Interscience, New York, NY, USA, 1988.

[126] A. Niederlinski. A heuristic approach to the design of linear multivariable interacting control systems. *Automatica*, 7(6):691–701, 1971.

[127] J. Nocedal and S. Wright. *Numerical Optimization*. Springer Verlag, 2nd edition, 2006.

[128] J. Oldenburg. *Logic-Based Modeling and Optimization of Discrete-Continuous Dynamic Systems*. PhD thesis, RWTH Aachen University, 2005.

[129] B. Pahor, Z. Kravanja, and N. Bedenik. Synthesis of reactor networks in overall process flowsheets within the multilevel MINLP approach. *Computers & Chemical Engineering*, 25:765–774, 2001.

[130] E. Panier and A. Tits. A globally convergent algorithm with adaptively refined discretization for semi-infinite optimization problems arising in engineering design. *IEEE Transactions on Automatic Control*, 34(8):903–908, 1989.

[131] A. Pereira, M. Costa, and E. Fernandes. Interior point filter method for semi-infinite programming problems. *Optimization*, 60(10-11, SI):1309–1338, 2011.

[132] L. Perko. *Differential Equations and Dynamical Systems*. Springer, New York, 2nd edition, 1996.

[133] A. Peschel, H. Freund, and K. Sundmacher. Methodology for the design of optimal chemical reactors based on the concept of elementary process functions. *Industrial & Engineering Chemistry Research*, 49:10535–10548, 2010.

https://doi.org/10.51202/9783186952035 - Generiert durch IP 216.73.216.36, am 18.01.2026, 21:11:22. © Urheberrechtlich geschützter Inhalt. Ohne gesonderte Erlaubnis ist jede urheberrechtliche Nutzung untersagt, insbesondere die Nutzung des Inhalts im Zusammenhang mit, für oder in KI-Systemen, KI-Modellen oder Generativen Sprachmodellen.

[134] L. Petzold. Differential/algebraic equations are not ODEs. *SIAM Journal on Scientific and Statistical Computing*, 3(3):367–384, 1982.

[135] E. Polak. On the use of consistent approximations in the solution of semi-infinite optimization and optimal control problems. *Mathematical Programming*, 62(1-3): 385–414, 1993.

[136] E. Polak and L. He. Rate-preserving discretization strategies for semi-infinite programming and optimal control. *SIAM Journal on Control and Optimization*, 30(3): 548–572, 1992.

[137] C. Price and I. Coope. Numerical experiments in semi-infinite programming. *Computational Optimization and Applications*, 6(2):169–189, 1996.

[138] A. Raghunathan and L. Biegler. An interior point method for mathematical programs with complementarity constraints (MPCCs). *SIAM Journal on Optimization*, 15(3): 720–750, 2005.

[139] D. Ralph and S. Wright. Some properties of regularization and penalization schemes for MPECs. *Optimization Methods and Software*, 19(5):527–556, 2004.

[140] R. Raman and I. Grossmann. Modeling and computational techniques for logic-based integer programming. *Computers & Chemical Engineering*, 18(7):563–578, 1994.

[141] J. Rawlings and J. Ekerdt. *Chemical Reactor Analysis and Design Fundamentals*. Nob Hill Publishing, 2002.

[142] R. Reemtsen. Discretization methods for the solution of semi-infinite programming problems. *Journal of Optimization Theory and Applications*, 71(1):85–103, 1991.

[143] R. Reemtsen and S. Görner. Numerical methods for semi-infinite programming: a survey. In *Semi-Infinite Programming*, pages 195–275. Kluwer Academic Publishers, Boston, Dordrecht, London, 1998.

[144] L. Ricardez-Sandoval, H. Budman, and P. Douglas. Integration of design and control for chemical processes: A review of the literature and some recent results. *Annual Reviews in Control*, 33-2:158–171, 2009.

[145] H. Rodrigues and M. Monteiro. Solving mathematical programs with complementarity constraints with nonlinear solvers. In A. Seeger, editor, *Recent Advances in Optimization*, volume 563 of *Lecture Notes in Economics and Mathematical Systems*, pages 415–424. Springer Berlin Heidelberg, 2006.

[146] W. Rooney and L. Biegler. Design for model parameter uncertainty using nonlinear confidence regions. *AIChE Journal*, 47:1794–1804, 2001.

[147] W. Rooney, B. Hausberger, L. Biegler, and D. Glasser. Convex attainable region projections for reactor network synthesis. *Computers & Chemical Engineering*, 24 (2-7):225–229, 2000.

https://doi.org/10.51202/9783186952035 - Generiert durch IP 216.73.216.36, am 18.01.2026, 21:11:22. © Urheberrechtlich geschützter Inhalt. Ohne gesonderte Erlaubnis ist jede urheberrechtliche Nutzung untersagt, insbesondere die Nutzung des Inhalts im Zusammenhang mit, für oder in KI-Systemen, KI-Modellen oder Generativen Sprachmodellen.

[148] V. Sakizlis, J. Perkins, and E. Pistikopoulos. Recent advances in optimization-based simultaneous process and control design. *Computer & Chemical Engineering*, 28(10): 2069–2086, 2004.

[149] H. Scheel and S. Scholtes. Mathematical programs with complementarity constraints: stationarity, optimality, and sensitivity. *Mathematics of Operations Research*, 25:1–22, 2000.

[150] G. Schembecker, T. Dröge, U. Westhaus, and K. Simmrock. READPERT - development, selection and design of chemical reactors. *Chemical Engineering and Processing: Process Intensification*, 34(3):317–322, 1995.

[151] W. Schiesser. *The Numerical Method of Lines: Integration of Partial Differential Equations*. Academic Press, 1991.

[152] S. Scholtes. Convergence properties of a regularization scheme for mathematical programs with complementarity constraints. *SIAM Journal on Optimization*, 11(4): 918–936, 2001.

[153] S. Scholtes and M. Stöhr. Exact penalization of mathematical programs with equilibrium constraints. *SIAM Journal on Control and Optimization*, 37(2):617–652, 1999.

[154] C. Schweiger and C. Floudas. Optimization framework for the synthesis of chemical reactor networks. *Industrial & Engineering Chemistry Research*, 38(3):744–766, 1999.

[155] H. Shaker and M. Komareji. Control configuration selection for multivariable nonlinear systems. *Industrial & Engineering Chemistry Research*, 51(25):8583–8587, 2012.

[156] M. Sharifzadeh. Integration of process design and control: a review. *Chemical Engineering Research and Design*, 91:2515–2549, 2013.

[157] D. Siljak. *Decentralized Control of Complex Systems*. Academic Press, Bosten, 1991.

[158] A. Singer and P. Barton. Global optimization with nonlinear ordinary differential equations. *Journal of global optimization*, 34(2):159–190, 2006.

[159] B. Srinivasan, S. Palanki, and D. Bonvin. Dynamic optimization of batch processes: I. characterization of the nominal solution. *Computers & Chemical Engineering*, 27 (1):1–26, 2003.

[160] S. Steffensen and M. Ulbrich. A new relaxation scheme for mathematical programs with equilibrium constraints. *SIAM Journal on Optimization*, 20(5):2504–2539, 2010.

[161] O. Stein. How to solve a semi-infinite optimization problem. *European Journal of Operational Research*, 223(2):312–320, 2012.

[162] O. Stein and G. Still. On optimality conditions for generalized semi-infinite programming problems. *Journal of Optimization Theory and Applications*, 104(2):443–458, 2000.

https://doi.org/10.51202/9783186952035 - Generiert durch IP 216.73.216.36, am 18.01.2026, 21:11:22. © Urheberrechtlich geschützter Inhalt. Ohne gesonderte Erlaubnis ist jede urheberrechtliche Nutzung untersagt, insbesondere die Nutzung des Inhalts im Zusammenhang mit, für oder in KI-Systemen, KI-Modellen oder Generativen Sprachmodellen.

[163] O. Stein and G. Still. On generalized semi-infinite optimization and bilevel optimization. *European Journal of Operational Research*, 142(3):444–462, 2002.

[164] O. Stein and G. Still. Solving semi-infinite optimization problems with interior point techniques. *SIAM Journal on Control and Optimization*, 42:769–788, 2003.

[165] O. Stein, J. Oldenburg, and W. Marquardt. Continuous reformulations of discrete-continuous optimization problems. *Computers & Chemical Engineering*, 28(10):1951–1966, 2004.

[166] G. Still. Generalized semi-infinite programming: theory and methods. *European Journal of Operational Research*, 119(2):301–313, 1999.

[167] G. Still. Discretization in semi-infinite programming: the rate of convergence. *Mathematical Programming*, 91(1):53–69, 2001.

[168] J. Sun. Eigenvalues and eigenvectors of a matrix dependent on several parameters. *Journal of Computational Mathematics*, 3(4):351–364, 1985.

[169] Y. Tanaka. A trust region method for semi-infinite programming problems. *International Journal of Systems Science*, 30(2):199–204, 1999.

[170] Y. Tanaka, M. Fukushima, and T. Ibaraki. A globally convergent SQP method for semi-infinite nonlinear optimization. *Journal of Computational and Applied Mathematics*, 23(2):141–153, 1988.

[171] M. Tawarmalani and N. Sahinidis. *Convexification and Global Optimization in Continuous and Mixed-Integer Nonlinear Programming: Theory, Algorithms, Software, and Applications*. Springer, 2002.

[172] M. Todd. Semidefinite optimization. *Acta Numerica*, 10:515–560, 2001.

[173] F. Trespalacios and I. Grossmann. Review of mixed-integer nonlinear and generalized disjunctive programming methods. *Chemie Ingenieur Technik*, 86(7, SI):991–1012, 2014.

[174] J. Vanbiervliet, B. Vandereycken, W. Michiels, S. Vandewalle, and M. Diehl. The smoothed spectral abscissa for robust stability optimization. *SIAM Journal on Optimization*, 20(1):156–171, 2009.

[175] M. Vandewal and B. Dejager. Control structure design: a survey. In *Proceedings of the 1995 American control conference*, pages 225–229, 1995.

[176] M. Vandewal and B. Dejager. A review of methods for input/output selection. *Automatica*, 37(4):487–510, 2001.

[177] A. Vecchietti and I. Grossmann. Modeling issues and implementation of language for disjunctive programming. *Computers & Chemical Engineering*, 24(9-10):2143 – 2155, 2000.

https://doi.org/10.51202/9783186952035 - Generiert durch IP 216.73.216.36, am 18.01.2026, 21:11:22. © Urheberrechtlich geschützter Inhalt. Ohne gesonderte Erlaubnis ist jede urheberrechtliche Nutzung untersagt, insbesondere die Nutzung des Inhalts im Zusammenhang mit, für oder in KI-Systemen, KI-Modellen oder Generativen Sprachmodellen.

[178] S. Veelken. *A New Relaxation Scheme for Mathematical Programs with Equilibrium Constraints: Theory an Numerical Experience*. PhD thesis, Fakultät für Mathematik, TU München,, 2009.

[179] P. Vega, R. Lamanna de Rocco, S. Revollar, and M. Francisco. Integrated design and control of chemical processes - part I: Revision and classification. *Computers & Chemical Engineering*, 71:602 – 617, 2014.

[180] F. Vzquez, J. Rückmann, O. Stein, and G. Still. Generalized semi-infinite programming: a tutorial. *Journal of Computational and Applied Mathematics*, 217(2):394 – 419, 2008.

[181] G. Watson. Globally convergent methods for semi-infinite programming. *BIT Numerical Mathematics*, 21(3):362–373, 1981.

[182] G. Watson. Lagrangian methods for semi-infinite programming problems. In E. Anderson and A. Philpott, editors, *Infinite Programming*, volume 259 of *Lecture Notes in Economics and Mathematical Systems*, pages 90–107. Springer Berlin Heidelberg, 1985.

[183] T. Westerlund and F. Pettersson. An extended cutting plane method for solving convex MINLP problems. *Computers & Chemical Engineering*, 19, Supplement 1: 131–136, 1995.

[184] M. Witcher and T. Macavoy. Interacting control systems - steady state and dynamic measurement of interaction. *ISA Transactions*, 16(3):35–41, 1977.

[185] H. Yamashita, H. Yabe, and K. Harada. A primaldual interior point method for nonlinear semidefinite programming. *Mathematical Programming*, 135(1-2):89–121, 2012.

[186] J. Ye. Optimality conditions for optimization problems with complementarity constraints. *SIAM Journal on Optimization*, 9(2):374–387, 1999.

[187] J. Ye. Necessary and sufficient optimality conditions for mathematical programs with equilibrium constraints. *Journal of Mathematical Analysis and Applications*, 307(1): 350 – 369, 2005.

[188] Z. Yuan, B. Chen, G. Sin, and R. Gani. State-of-the art and progress in the optimization-based simultaneous design and control for chemical processes. *AIChE Journal*, 58:1640–1659, 2012.

[189] X. Zhao and W. Marquardt. Reactor network design with guaranteed robust stability. *Computers & Chemical Engineering*, 86:75–89, 2016.

[190] X. Zhao and W. Marquardt. Closed-loop reactor network synthesis with guaranteed robustness. *Computers & Chemical Engineering, accepted*, 2017.

[191] Y. Zhao and M. Stadtherr. Rigorous global optimization for dynamic systems subject to inequality path constraints. *Industrial & Engineering Chemistry Research*, 50(22): 12678–12693, 2011.

https://doi.org/10.51202/9783186952035 - Generiert durch IP 216.73.216.36, am 18.01.2026, 21:11:22. © Urheberrechtlich geschützter Inhalt. Ohne gesonderte Erlaubnis ist jede urheberrechtliche Nutzung untersagt, insbesondere die Nutzung des Inhalts im Zusammenhang mit, für oder in KI-Systemen, KI-Modellen oder Generativen Sprachmodellen.

[192] Z. Zhu, Z. Luo, and J. Zeng. A new smoothing technique for mathematical programs with equilibrium constraints. *Applied Mathematics and Mechanics*, 28(10):1407–1414, 2007.

https://doi.org/10.51202/9783186952035 - Generiert durch IP 216.73.216.36, am 18.01.2026, 21:11:22. © Urheberrechtlich geschützter Inhalt. Ohne gesonderte Erlaubnis ist jede urheberrechtliche Nutzung untersagt, insbesondere die Nutzung des Inhalts im Zusammenhang mit, für oder in KI-Systemen, KI-Modellen oder Generativen Sprachmodellen.

https://doi.org/10.51202/9783186952035 - Generiert durch IP 216.73.216.36, am 18.01.2026, 21:11:22. © Urheberrechtlich geschützter Inhalt. Ohne gesonderte
Erlaubnis ist jede urheberrechtliche Nutzung untersagt, insbesondere die Nutzung des Inhalts im Zusammenhang mit, für oder in KI-Systemen, KI-Modellen oder Generativen Sprachmodellen.

Curriculum Vitae

Xiao Zhao

Aug. 16, 1983	Born in Xinjiang, China
1990-2002	Ürümqi Shihua Highschool
2002-2006	Bachelor Study of Industrial Engineering, Tsinghua University
2006-2009	Double Master study of Management Science and Engineering, Tsinghua University
2006-2009	Double Master study of Production System Engineering, RWTH Aachen
2009-2014	PhD study at Lehrstuhl für Prozesstechnik, RWTH Aachen
since 2014	Research assistant at Forschungszentrum Jülich GmbH

https://doi.org/10.51202/9783186952035 - Generiert durch IP 216.73.216.36, am 18.01.2026, 21:11:22. © Urheberrechtlich geschützter Inhalt. Ohne gesonderte
Erlaubnis ist jede urheberrechtliche Nutzung untersagt, insbesondere die Nutzung des Inhalts im Zusammenhang mit, für oder in KI-Systemen, KI-Modellen oder Generativen Sprachmodellen.

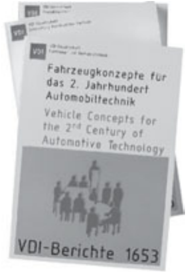
https://doi.org/10.51202/9783186952035 - Generiert durch IP 216.73.216.36, am 18.01.2026, 21:11:22. © Urheberrechtlich geschützter Inhalt. Ohne gesonderte
Erlaubnis ist jede urheberrechtliche Nutzung untersagt, insbesondere die Nutzung des Inhalts im Zusammenhang mit, für oder in KI-Systemen, KI-Modellen oder Generativen Sprachmodellen.

Online-Buchshop für Ingenieure

VDI nachrichten

Online-Shops

Fachliteratur und mehr - jetzt bequem online recherchieren & bestellen unter: www.vdi-nachrichten.com/Der-Shop-im-Ueberblick


Fahrzeugkonzepte für das 2. Jahrhundert Automobiltechnik
Vehicle Concepts for the 2nd Century of Automotive Technology
VDI-Berichte 1653

Täglich aktualisiert: Neuerscheinungen VDI-Schriftenreihen

Fortschritt-Berichte VDI
Weiterentwicklung der Endfunktionsprüfung von Verbrennungskraftmaschinen in der Großserienproduktion

BUCHSHOP

Im Buchshop von vdi-nachrichten.com finden Ingenieure und Techniker ein speziell auf sie zugeschnittenes, umfassendes Literaturangebot.

Mit der komfortablen Schnellsuche werden Sie in den VDI-Schriftenreihen und im Verzeichnis lieferbarer Bücher unter 1.000.000 Titeln garantiert fündig.

Im Buchshop stehen für Sie bereit:

VDI-Berichte und die Reihe **Kunststofftechnik:**

Berichte nationaler und internationaler technischer Fachtagungen der VDI-Fachgliederungen

Fortschritt-Berichte VDI:

Dissertationen, Habilitationen und Forschungsberichte aus sämtlichen ingenieurwissenschaftlichen Fachrichtungen

Newsletter „Neuerscheinungen":

Kostenfreie Infos zu aktuellen Titeln der VDI-Schriftenreihen bequem per E-Mail

Autoren-Service:

Umfassende Betreuung bei der Veröffentlichung Ihrer Arbeit in der Reihe Fortschritt-Berichte VDI

Buch- und Medien-Service:

Beschaffung aller am Markt verfügbaren Zeitschriften, Zeitungen, Fortsetzungsreihen, Handbücher, Technische Regelwerke, elektronische Medien und vieles mehr – einzeln oder im Abo und mit weltweitem Lieferservice

VDI nachrichten | **BUCHSHOP** **www.vdi-nachrichten.com/Der-Shop-im-Ueberblick**

https://doi.org/10.51202/9783186952035 - Generiert durch IP 216.73.216.36, am 18.01.2026, 21:11:22. © Urheberrechtlich geschützter Inhalt. Ohne gesonderte Erlaubnis ist jede urheberrechtliche Nutzung untersagt, insbesondere die Nutzung des Inhalts im Zusammenhang mit, für oder in KI-Systemen, KI-Modellen oder Generativen Sprachmodellen.

Die Reihen der Fortschritt-Berichte VDI:

1 Konstruktionstechnik/Maschinenelemente
2 Fertigungstechnik
3 Verfahrenstechnik
4 Bauingenieurwesen
5 Grund- und Werkstoffe/Kunststoffe
6 Energietechnik
7 Strömungstechnik
8 Mess-, Steuerungs- und Regelungstechnik
9 Elektronik/Mikro- und Nanotechnik
10 Informatik/Kommunikation
11 Schwingungstechnik
12 Verkehrstechnik/Fahrzeugtechnik
13 Fördertechnik/Logistik
14 Landtechnik/Lebensmitteltechnik
15 Umwelttechnik
16 Technik und Wirtschaft
17 Biotechnik/Medizintechnik
18 Mechanik/Bruchmechanik
19 Wärmetechnik/Kältetechnik
20 Rechnerunterstützte Verfahren (CAD, CAM, CAE CAQ, CIM ...)
21 Elektrotechnik
22 Mensch-Maschine-Systeme
23 Technische Gebäudeausrüstung

ISBN 978-3-18-3**95203**-8

https://doi.org/10.51202/9783186952035 - Generiert durch IP 216.73.216.36, am 18.01.2026, 21:11:22. © Urheberrechtlich geschützter Inhalt. Ohne gesonderte
Erlaubnis ist jede urheberrechtliche Nutzung untersagt, insbesondere die Nutzung des Inhalts im Zusammenhang mit, für oder in KI-Systemen, KI-Modellen oder Generativen Sprachmodellen.