A Game-Theoretic Approach
For Multi-loop Control System Design

Inauguraldissertation
zur Erlangung des akademischen Grades
eines Doktors der Naturwissenschaften
der Universität Mannheim

vorgelegt von

Dipl.-Inf. Andrea Richter
aus Mannheim

Mannheim, 2011

Dekan:	Professor Dr. H. J. Müller, Universität Mannheim
Referent:	Professor Dr. E. Badreddin, Universität Heidelberg
Korreferent:	Professor Dr. C. de Prada Moraga, Universidad de Valladolid

Tag der mündlichen Prüfung: 23. Mai 2013

Abstract

A new approach is developed in the present thesis, to tune PI-Controllers in a Multi-loop Control System, regarding to certain control loop properties, constraints and requirements.
Several control loops in a Multi-loop Control System interact to a greater or lesser extend, depending on the control loop structure. This behavior could yield to a degradation of one loop requirement due to an optimization of a second control loop in tuning it's controllers according to another requirement. This is a conflict situation, which could not be avoided, but improved. A consideration of one or more requirements in a Multi-loop Control System is known as a multi-objective optimization problem. To solve such a kind of Multi-objective problem, a new approach is developed that supports finding a fair trade-off from the point of view of all requirements, criteria and each control loop of a Multi-loop Control System. The developed approach uses tools from game theory, which could be applied meaningfully, to describe and solve conflict situations as described within several control loop requirements and criteria. Assistant steps of the approach are that the conflict situation, respectively the Multi-objective optimization problem is structured and described as a game, first. The belonging requirements as well as constraints are formalized mathematically in the next step. Finally, game theory provides several solution concepts to calculate a fair trade-off, which is the solution to the game.
In the wide research field of game theory, the structure of information plays a decisive role. This fact leads to a second application in the control theory of the developed approach. Using different information structures of a game, respectively a Multi-loop Control System, leads to a change of the players' strategy sets. This has a bearing on the final solution of the game. Through this game-theoretic point of view, multiple Multi-loop Control System structures could be analyzed and compared for one and the same control theoretic problem.

Kurzbeschreibung

In der vorliegenden Arbeit wird ein neuer Ansatz zur Einstellung der Parameter eines PI Reglers in einem mehrschleifigen Regelsystem entwickelt. Dieser Ansatz berücksichtigt sowohl bestimmte Eigenschaften der Regelschleifen, wie auch Einschränkungen und Anforderungen der Regelschleifen. In einem mehrschleifigen Regelsystem interagieren die Regelschleifen, je nach Struktur der Regelschleifen, mehr oder weniger stark miteinander. Dieses Verhalten kann zu einer Herabstufung einer Regelschleifenanforderung führen, wenn bei der Parametereinstellung die Optimierung einer zweiten Regelschleifenanforderung im Vordergrund steht. Die daraus entstehende Konfliktsituation kann nur selten verhindert werden. Das Regelverhalten kann jedoch in Bezug auf mehrere Anforderungen oder Beschränkungen der Regelschleifen deutlich verbessert werden. Die Betrachtung von ein oder mehreren Anforderungen in einem Mehrschleifenregelsystem wird auch als multikriterielles Optimierungsproblem bezeichnet.
Das Ziel des hier entwickelten Ansatzes ist es, eine faire Kompromisslösung aus Sicht aller Regelanforderungen und Beschränkungen zu bekommen. Hierbei werden auch die einzelnen Regelschleifen des Systems miteinbezogen. Der hier entwickelte Ansatz stützt sich auf Werkzeuge der Spieltheorie, welche angewendet werden können, um die bestehende Konfliktsituation aus unterschiedlichen Regelkreisanforderungen und deren Beschränkungen zu beschreiben und zu lösen. Ein hilfreicher erster Schritt des Ansatzes ist es, die Konfliktsituation, beziehungsweise das multikriterielle Optimierungsproblem zu strukturieren und als Spiel zu beschreiben. In einem weiteren Schritt werden die zugehörigen Anforderungen, sowie die Einschränkungen mathematisch beschrieben. Mittels unterschiedlicher Lösungskonzepte aus der Spieltheorie, ergibt sich die Lösung des Spiels aus der Berechnung eines fairen Kompromisses. Das Ergebnis ist ein Parametersatz für die Regler, welche alle Anforderungen gleichberechigt erfüllt.
Auf dem weiten Forschungsgebiet der Spieltheorie spielt die vorliegende Informationsstruktur eine entscheidende Rolle. Diese Tatsache führt zu einer weiteren Anwendung des in dieser Arbeit entwickelten Ansatzes: die Nutzung von verschiedenen möglichen Informationsstrukturen eines Spiels, beziehungsweise eines Merhschleifenregelsystems, führt zu unterschiedlichen Strategiemengen der Spieler. Diese Strategiemengen haben wiederum einen Einfluß auf die Lösung des Spiels.

Durch diese spieltheoretische Betrachtung können unterschiedliche Strukturen von Mehrschleifenregelsystemen für ein und dasselbe regelungstechnische Problem analysiert und verglichen werden.

Contents

List of Figures

List of Tables

List of Abbreviations

C	Permeate conductivity, p. 66
CL	Characteristic-Locus, p. 14
DM	Decision maker, p. 54
DNA	Direct Nyquist Array, p. 14
E	Egalitarian solution, p. 28
F	Permeate flux, p. 66
GA	Genetic algorithm, p. 8
GEATbx	Genetic evolutionary algorithm toolbox, p. 38
INA	Inverse Nyquist Array, p. 14
ISE	Integral square error, p. 8
ISTSE	Integral of square time weighted square error, p. 48
ITSE	Integral of time weighted square error, p. 48
KS	Kalai-Smorodinsky solution, p. 28
MIMO	Multi-input/multi-output system, p. 1
MOGA	Multi-objective genetic algorithm, p. 36
MOO	Multi-objective optimization, p. 2
MPC	Model predictive control, p. 13
NB	Nach bargaining solution, p. 27
NBI	Normal boundary intersection, p. 36

NSGA-II	Non-dominated sorting genetic algorithm II, p. 36
P	Transmembrane pressure, p. 66
pH	Feed pH, p. 66
PI	Proportional-integral, p. 7
PID	Proportional-integral-derivative, p. 7
PRG	Partial Relative Gain, p. 7
PRGA	Performance Relative Gain Array, p. 7
PR	Proportional solution, p. 28
RGA	Relative Gain Array, p. 7
RO	Reverse osmosis, p. 65
SISO	Single-input/single-output system, p. 5
TITO	Two-input/two-output system, p. 66
U	Utopia point, p. 20

List of Symbols

$(\cdot)^*$	Optimal solution, p. 26
$(\cdot)_v$	Stochastic disturbance, p. 54
α	Decision variable, p. 30
β	Weighting factor, p. 32
γ, δ	Parameters, p. 15
$\boldsymbol{\chi}$	Parameter vector for the genetic algorithm, p. 73
Δ	Uncertainty block, p. 92
Δu	Measure of change in control signal, discrete difference, p. 49
λ	Lagrange multiplier, p. 74
$\mathcal{L}$	Laplace-transformed, p. 43
$\mathcal{Z}$	$\mathcal{Z}$-transformed, p. 46
$\mathbb{R}^+$	Field of non-negative real numbers, p. 21
$\mathbb{R}^n$	n-dimensional field of real numbers, p. 34
μ	Structured singular value, p. 52
fp	Feasible point, p. 34
σ	Singular value analysis, p. 52
τ	Time constant, p. 43
$\boldsymbol{\alpha}$	Vector of decision variables, p. 32
^{T}	Transposed of a vector, p. 33

A Denumerator polynomial of a process transfer function, p. 41

$\mathcal{A}$ Set of decision vectors, p. 30

a Coefficients of the differential or difference equation of the transfer function's denumerator, p. 41

B Numerator polynomial of a process transfer function, p. 41

b Coefficients of the differential or difference equation of the transfer function's numerator, p. 41

c Control signal function of time, p. 43

C Controller, controller transfer function p. 6

C^* Denumerator polynomial of a filter transfer function, p. 100

$(\cdot)_{cf}$ Indexing which kind of cost function is used, p. 48

d Disagreement point, p. 22

e Error signal function of time, p. 43

$\mathcal{E}$ Euclidian space, p. 33

E Error signal polynomial, p. 42

ϕ, ϵ, fnb Functions, describing a bargaining game, p. 28

f function, describing error evolution, p. 43

g function, describing the cost, p. 43

η Inequality constraint functions, p. 33

h Equality constraint functions, p. 33

G Process transfer function, p. 5

H_∞ Norm for Optimization , p. 8

H_2 Norm for Optimization, p. 8

i, j Index i, j, indexing the players, p. 6

$\mathcal{I}$ Information set, information structure, p. 45

I Unit matrix, p. 52

ι, l Indexing (in)equality constraint functions, p. 33

J	Cost function, p. 21
J_p	Perfomance index, p. 115
k	Discrete time, p. 40
k_0	Starting time, p. 44
k_1, k_2	amplification factors, p. 16
K	End time, p. 44
K_P	Proportional parameter, p. 42
K_I	Integral parameter, p. 42
K_T	Reset time, p. 42
L	Limit, p. 50
M	Transfer function matrix of a linear system, p. 52
m, w	Arbitrary real numbers, p. 28
max	Maximization, p. 30
min	Minimization, p. 30
n, n_A, n_B	Order of the system, order of polynomial A, order of polynomial B, p. 45
N	Number of players, number of control signals, p. 13
$N(\mathcal{S})$	Nash bargaining solution value of a feasible set $\mathcal{S}$, p. 27
P	Denumerator polynomial of a controller transfer function, p. 42
$\mathcal{P}, \mathcal{Q}, \mathcal{R}, \mathcal{S}$	Feasible sets, p. 22
Q	Numerator polynomial of a controller transfer function, p. 42
r	Set point input signal, p. 6
R	Refinement factor, p. 116
S	Coalition, p. 22
s	Complex frequence, p. 15
t	Continuous time, p. 40

t_0	Starting time, p. 40
T	End time, p. 40
t_f	Simulation end time, p. 115
T_0	Sample time, p. 102
U	Control signal polynomial, input polynomial, p. 41
$\mathcal{U}$	Action space, strategy space p. 20
u	Control variable, game strategy, p. 5
Y	Output polynomial, p. 41
y	Output variable, p. 5
z	Complex variable used in z-transform, p. 45
ψ	Number of decision variables α , p. 33
υ	Number of inequality constraint functions, p. 33
ξ	Number of equality constraint functions, p. 33
$\dot{(\cdot)}, \ddot{(\cdot)}, (\cdot)^{(n)}$	First, second, $n-$th derivative, p. 49
x_a, x_b	Arbitrary real numbers, p. 30

Chapter 1

Introduction

1.1 Motivation

Today, the use of complex large scale systems in industry is well-established. Due to their complexity, such systems normally operate with multiple inputs and multiple outputs. For this reason they are also called multi-input/multi-output (MIMO) systems. The handling of MIMO systems, including their high complexity, often requires the implementation of sophisticated control structures. A suitable method is a multi-loop control. According to (Johnson u. Moradi, 2005), the use of multi-loop controls is widespread in the industry, due to the simplicity of implementation as well as the possibility of manual tuning. However, just in the simplicity of implementation and the possibility of manual tuning, the consideration of multi-loop interactions are neglected. Opening or closing one loop could result in a change of the dynamics in the other control loops. Concerning this matter, they demonstrate the main disadvantage in the use of multi-loop controls.
Though, not only the interactions of the several loops play a decisive role during the control system design of MIMO systems, but also the different demands that are made on the system have to be involved. (Compare (Bernard u. a.), which are displayed in Fig. 1.1)

The requirements on the control system design listed in Fig. 1.1 are several, but partial conflicting criteria. For example, a low control effort is often achieved at the expense of robustness and a fast reference tracking with minimal deviation which are conflicting criteria. The problem in achieving all requirements just

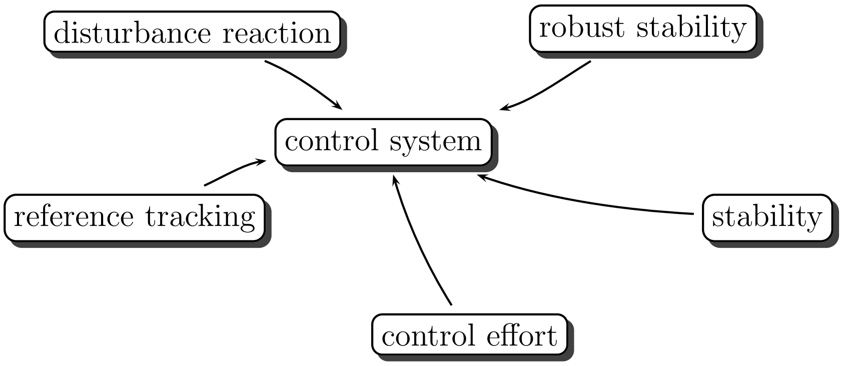

Figure 1.1. Requirements on control system.

for a single-loop control system is well treated in the literature, see (Andersson, 2000), (Bernard, 2005), (Elia u. Dahleh, 1997), (Herreros u. a., 1999), (Hutauruk u. Brown, 2005), (Kawabe u. Tagami, 1999), (Kookos u. a., 1999) and (Tagami u. a., 2004). The main contribution of the aforementioned references is to find a satisfying trade off within the requirements in treating it as a multi-objective optimization problem (MOO).

If the control system is a multi-loop, the requirements of Fig. 1.1 has to be achieved for every control loop as well as for the total system. For multi-loop control systems, it is difficult to set up a unique performance index that satisfies the specifications for all the control loops.

Hence, a performance index for each control loop has to be defined. If the system dynamic allows for it, each control loop can be tuned separately according to the performance indices. This is the standard method developed in the past, where much research effort was dedicated to obtain decoupling methods. If the system dynamics do not allow decoupling and a separated tuning of each controller, then a joint optimization has to be applied.

According to (Johnson u. Moradi, 2005), achieving a satisfactory loop performance for multi-loop systems represents a great challenge in the area of control system design. This results in the existence of only a few powerful tools, applicable to such systems.

The focus of this work is to develop an efficient approach to perform a joint optimization of all requirements with respect to loop interactions in MIMO systems.

This approach requires some kind of strategic decision making for systems with interacting and conflicting objectives.
One research field with it's origin in the early 1920s, dealing with strategic decision making is game theory. Game theory is an instrument to analyse the conflicting situations of decision making, where more than one decision maker, also called player, try to follow their individual and often conflicting objectives.
A classification of different types of games that are related to mathematical concepts and are already known in the literature is given in Tab.1.1.

Table 1.1. Classification of different game types, related to the corresponding mathematical concept.

	One player	Many players
Static	Mathematical programming	Static game theory
Dynamic	Optimal control	Dynamic game theory

According to Tab.1.1, game-theoretic concepts are addressed if many players participate, in contrast to the mathematical programming and the optimal control, used as tools for one player games.
The approach, that is developed in this work, has it's origin in game theory. The game-theoretic consideration of a problem represents the situation of decision making explicitly and supports it's formalisation in assisting the development of optimal solutions. A game-theoretic problem is solved first in sorting the available information, like:

- Who are the participants?
- Which participant has which information?
- Which strategies are used?
- What is the outcome for each participant according to their behavior (strategy selection)?
- According to which rules is the behavior of the participants?

Alternative strategy selections are simulated and optimized with the goal of satisfying the outcome of each participant. Actually, game theory detects negative incentives, that could exist in the system. Game theory models the mathematical relations around the strategic behavior in situations of competition and conflict.
According to the aforementioned essential subject of game theory - the analysis of a strategic decision process - it is predestinated in this work as a modeling tool.
With the aid of concepts and formalisations that have been defined in game theory, a game-theoretic approach is developed. Here, the control system design is described as a game and the solution of the game provides a set of solutions. This satisfies specified demands and include the handling with sometimes unavoidable interactions.
Considering the control system design as a game, the controllers are viewed as players, while the different objectives of the different players are incorporated as well as their interactions, caused by the strategy choice of each player.
In the core of the game-theoretic approach, the control system design remains a multi-objective optimization, where multiple defined cost functions, describing the objectives or requirements on the control loops of the system, are optimized simultaneously. In it's development, based on game theory, more possibilities for manipulation and variances in the implementation of the control system design problem should be provided. For instance, an essential component of game theory is the information structure with a major influence on the description, course and solution of the game. Players could be part of a cooperative or a non-cooperative game, they could form coalitions, or act alone, which all leads to different game descriptions, rules and certainly different solutions.
The perspective on the information structure provides a second application of the developed approach: from game-theoretic view, a change in the information structure could cause a change in the strategy spaces, available for the players, resulting in another solution of the game. Thus, in the face of different information distribution in a game, diverse resulting control structures could be analysed and compared with each other.

1.2 Problem statement

The focus of this work is to develop a new approach for the control system design of multi-loop control structures using a game-theoretic background. An example of a MIMO system with three inputs, three outputs and five sub-processes is displayed in Fig. 1.2. The three inputs u_1, u_2 and u_3 are the control signals for the five processes together. For example process G_5 is controlled by the sum of the inputs u_2 and u_3. The same applies for the outputs y_1, y_2 and y_3. They are composed of the sum of G_1, G_2 and G_3 and the sum of G_4 and G_5, respectively.
A corresponding possibility of a multi-loop control structure for the control of the MIMO system in Fig. 1.2 is shown in Fig. 1.3.

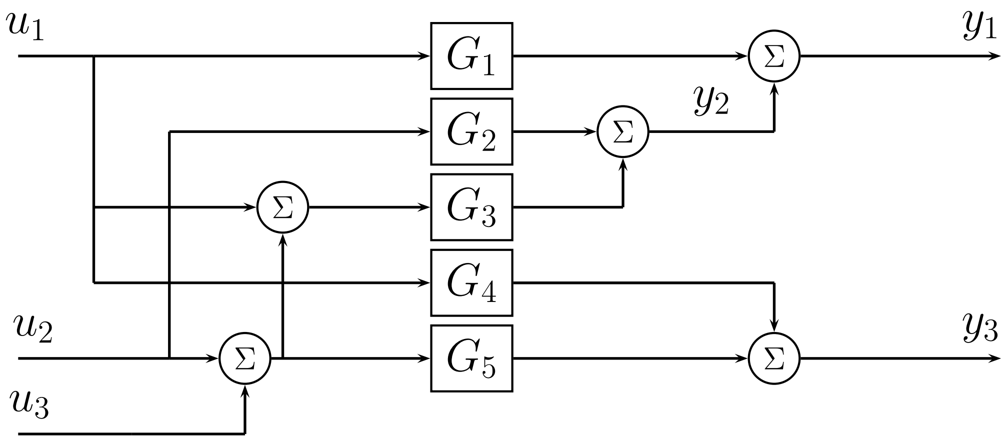

Figure 1.2. Example of a MIMO system with three inputs u_1, u_2, u_3 and two outputs y_1 and y_2.

The control structure of Fig. 1.3 consists of more than one control loop as well as more than one single-input/single-output (SISO) controller.
The special characteristic of such structures is the interaction of the different control loops. Assuming, a controller is located in each control loop, as it is shown in Fig. 1.3, and the controller parameters have to be tuned. One method, which is widespread in the literature, is to consider each control loop separately and to tune the controller parameters, see (Qiang Xiong u. He). For example, with the classical method of Ziegler-Nichols, developed in 1942. Thereby, all interactions are neglected.

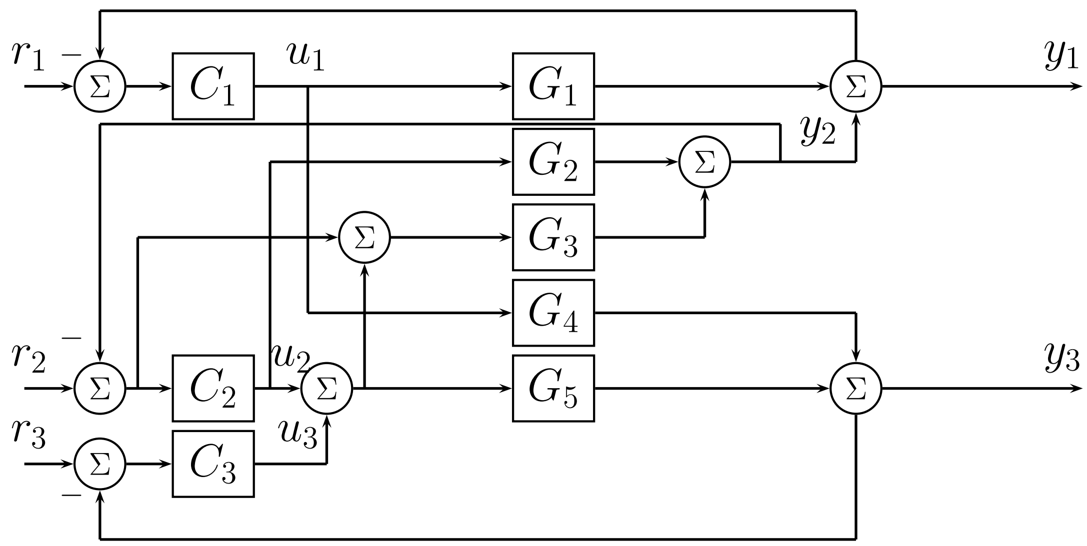

Figure 1.3. Example of a MIMO system with three inputs u_1, u_2, u_3 and three outputs y_1, y_2 and y_3.

A further method is to pretreat the MIMO system using decoupling techniques, which is described in detail in (Lunze, 2004). The intention of a decoupler is to eliminate the effect of interaction mathematically by transforming the process matrix into a diagonal matrix.
However, in using multi-loop controls, often neglected is the strength of coupling different variables (typical control methods to measure loop interactions are the Relative Gain Array (RGA), see (Bristol, 1966) or Gramian based methods, see (Conley u. Salgado, 2000)). In addition, the decoupling approach is limited through basics of control theory, since the method could translate zeros to poles and unstable decoupling elements may result. In the case of using a decoupling approximation algorithm, two questions remain:

1. How large is the additional effort of using an approximation algorithm?

2. How exact or precise is the approximation?

Of course, the question if other adverse effects that arise could also be posed.
Thus, a qualified method is demanded, that considers these interactions during control system design and achieve requirements of Fig. 1.1 at the same time.
In considering the problem of the control system design for MIMO systems as a game, where the controllers are viewed as players, the different objectives of

the different players are incorporated as well as their interactions caused by the strategy choice of each player.

1.3 State of the art

A detailed literature research, for example in (Brosilow u. Joseph, 1999), pointed out that there is no satisfying method for simultaneous tuning of several controllers that significantly improved performance compared to a single loop controller.
According to (Johnson u. Moradi, 2005), the disadvantages of using multi-loop proportional-integral (PI) and proportional-integral-derivative (PID) controllers are the lack of loop interaction consideration and the existence of few powerful tools for its design.
A typical control method to measure process interactions is the Relative Gain Array (RGA), developed by (Bristol, 1966). The RGA provides a measurement method of best pairing for controlled and manipulated variables. Variations, based on RGA are the Partial Relative Gain Array (PRG), see (Häggblom, 1997), or the Performance Relative Gain Array (PRGA), see (Hovd u. Skogestad, 1992). Another way of loop interaction measurement is based on Gramians, which was presented first in (Conley u. Salgado, 2000) and (Salgado u. Conlea, 2004) using Participation Matrices (PM). In such measurements, the whole frequency range is taken into account with one single measurement. In (Rosenbrock, 1974), Gershgorin circles are used to analyze the loop interactions in multivariable systems.
Standard techniques for controller tuning of multi-loop control systems assume that the control loops can be adjusted individually by loop decoupling, thereby neglecting the interactions of the different control loops.
As mentioned in Section 1.1, multi-objective optimization is a principle component of the game-theoretic approach, developed in this work. The idea of multi-objective optimization was first introduced in the field of automatic control in (Lin, 1976) as a method to deal with many incommensurable as well as incompatible objectives. The design of controllers often entails conflict situations of many criteria, such as control energy, tracking performance or robustness. This situation is described in (Hutauruk u. Brown, 2005) as a MOO problem whereas the solution of this problem is presented for the design of PI and PID controllers. A further controller

design approach considering multiple objectives is proposed in (Elia u. Dahleh, 1997). A multi-objective optimization approach to design robust controllers using genetic algorithms (GA) is described in (Herreros u. a., 1999). The design problem of a robust PID controller with two degrees of freedom based on the partial model matching approach is treated in (Kawabe u. Tagami, 1999). Herein, the design problem is formulated as a two objective minimax optimization problem and a new genetic algorithm using a Pareto partitioning method for the controller design problem is shown. In order for solving the multi-objective optimization problem for PI and PID controller tuning, a simplified goal-attainment formulation is used in (Kookos u. a., 1999). A design procedure for tuning PID controller parameters to achieve a mixed H_2/H_∞ optimal performance using genetic algorithms is described in (Calistru, 1999). The H^∞-optimal control problem minimizes the maximum of the H^∞-norm of a transfer function matrix, which is the maximum of it's largest singular value over all frequencies. A two objective optimization problem of a robust I-PD controller design is solved in (Kawabe u. Tagami, 1999) using a genetic algorithm. The proposed controller design method is based on the generalized Integral Square Error (ISE) criterion while the genetic algorithm is applied to optimize the problem with multiple search property as an advantage, compared to other MOO solution tools.

The combination of multi-objective optimization and control system design is outlined in (Liu u. a., 2002), where one PI controller for a MIMO system was designed using multi-objective optimization. During the design of the controller, constraints on outputs and inputs are considered. These are formulated as a performance function criteria. However, more specific requirements on the closed loop system are not defined.

Surveys of MOO in engineering design are proposed in (Andersson, 2000), (Marler u. Arora, 2004) and (Osyczka, 1985). An updated survey of MOO applications for control systems is presented in (Gambier, 2007).

In contrast, using game theory as tool for control system design is known in the literature, indeed, but not in the dimension compared to multi-objective optimization. The application of game theory as tool for control system design is already presented in the literature.

First steps to use a dynamic (differential) game-theoretic framework for a worst-

case controller design for linear plants were made by George Zames in the early 1980's (Zames, 1981). However, earlier works on worst-case controller designs exists starting in the 1950's. During this period, the research field of dynamic game theory was actually at the beginning stages and not yet applicable.
The worst-case controller design for linear plants subject to unknown additive disturbances and plant uncertainties is originally known as a H^{∞}-optimal control problem.
The H^{∞}-optimal control problem is in fact a minimax optimization problem, it can be viewed as a zero-sum game with the controller as the minimizing player and the disturbance as the maximizing player. The developed design of (Zames, 1981), that minimizes a given performance index under worst possible disturbances or parameter variations is summarized and prepared in (Basar u. P.Bernhard, 1995). The relationship between a game-theoretic controller and a H^{∞}-controller is demonstrated in (Rhee u. Speyer, 1989).
Finally, a well known approach for hybrid controller design in a game-theoretic framework is published in the years between 1995 and 2000 by Lygeros, Tomlin, Godbole and Sastry, see (Tomlin u. a., 2000), (Lygeros u. a., 1995), (Lygeros u. a., 1996) and (Lygeros u. a., 1997). In their work, the control problem is given as a hybrid, large scale, multi agent system with the objective to develop a hybrid, hierarchical controller design.

1.4 Main Contribution

The main contribution of this thesis is allocated on two main aspects. First, a game-theoretic approach is developed for the control system design of multi-loop control systems. Second, a topological analysis of a multi-loop control structure is proposed on the basis of the developed game-theoretic approach.
Concerning the first aspect, game theory is used to model and solve the problem of control system design in multi-loop systems. The developed approach is organized as follows:

(I) A game description of a multi-loop control system

(II) Cost function and constraint set up, considering loop interactions and different system requirements

(III) A Pareto-optimal set as solution set of the game, obtained through MOO, which is solved using a GA

(IV) A solution concept for the final solution, chosen from the Pareto-optimal set

The game-theoretic description of the control system design for a multi-loop system mainly contains players, strategy sets and cost functions, compare item (I) and (II). The simultaneous optimization of the players' cost functions provides a set of solutions, called the Pareto-optimal set, while each single solution of the solution set states for an efficient compromise within the different cost functions, see item (III). The new approach is developed and successfully applied for the continuous case as well as for the discrete case ((Wellenreuther u. a., 2006b), (Wellenreuther u. a., 2006a)).
A simultaneous consideration of specified system requirements, like a fast reference tracking with low deviation, low control effort, robustness and a fast disturbance reaction is implemented using corresponding cost functions in ((Wellenreuther u. a., 2007), (Wellenreuther u. a., 2008b), (Wellenreuther u. a., 2008a)). The final solution of the control system design is based on a game-theoretic solution concept concerning item IV). Three main solution concepts, according to a special control system design game are implemented and compared, where the game solution provides the controller parameters for the particular controllers.
The second main aspect corresponds to a topological analysis, based on the game-theoretical approach. The proposed topological analysis profits from an essential component in a game, the information. The information concerning which player knows what and when is given in the information structure of the game. Different information structures from the game-theoretic view could lead to different control system topologies for one single MIMO system from the control-theoretic view. This fact is used, to analyse different possible control structures and compare them, according to their information structure ((Wellenreuther u. a., 2008c), (Wellenreuther u. a., 2008b)).

1.5 Outline of the thesis

Chapter 2: The present work is based on control-theoretic concepts as well as game-theoretic concepts. To be able to follow the developed approach, based on both research areas, basics concerning MIMO systems, their control and required fundamentals in game theory are presented.

Chapter 3: Adapted from the preliminaries in Chapter 2, the developed approach for the control system design of multi-loop systems, using game theory is proposed in Chapter 3, for the continuous case as differential game as well as for the discrete case as difference game. Additionally, several cost functions and constraints, corresponding to different system requirements, are formalized.

Chapter 4: A topological analysis of the control structure is proposed in considering different possible information structures in the game description of the control system design. This leads to different possible control structures for one original MIMO system.

Chapter 5: The proposed game-theoretical approach for the control system design is described in detail and applied to an example of a continuous MIMO system, considering the system requirements on reference tracking, low control effort and the robust stability. Concerning the requirement on low control effort, four different possibilities are implemented and compared.

Chapter 6: The discrete representation of a MIMO example provides the basis for the control system design subject to a fast reference tracking with low deviation. The control system design is formulated using the implementation of three different cost functions and comparing their resulting system behaviors.

Chapter 7: The topological analysis of a control structure is applied on an example, while the differences as well as the advantages and disadvantages of the different structures are identified. In a second step, constraints on the strategy sets are added and their effects on the Pareto-optimal sets are studied.

Chapter 8: In order to refine the comparison of the different game implementations, cost indices are calculated during the simulations. On the basis of the cost indices, a refinement factor is computed for each game to evaluate its performance.

Chapter 9: The work is concluded with a summary and an outlook. These contain a condensed version of the proposed approach, a critical consideration of the approach as well as still open questions and further resulting work.

Chapter 2

Preliminaries

2.1 Control of Multi-variable Processes

Multi-variable processes are complex systems with many mutual interacting input and output variables. Usually, the control variables as well as the controlled process variables are coupled directly. Furthermore, one control signal u_i with $i = 1, .., N$ with N representing the number of control signals, affects multiple controlled process variables.
In the literature, two ways for the control of multi-variable processes are distinguished:

- Multi-variable control
- Decentralized or multi-loop control

Below, the multi-variable control concept is briefly summarized, while the decentralized control concept is described in more details. Their distinguishing features are pointed out according to (Ghavipanjeh, 2006) and (Skogestad, 2003).

2.1.1 Multi-variable control

In multi-variable controlled systems, MIMO systems are controlled with one MIMO controller in the size of the multi-variable application. If explicit constraint handling is applied, as it is usually the case, the control method is known as model predictive control (MPC) with a smooth movement between the changing active constraints. Other multivariable control design techniques are known as Nyquist-array methods, like the Direct Nyquist Arrays (DNA) and the Inverse Nyquist-Array (INA), where the transfer function matrix shall be made diagonal dominant by the use of a compensator matrix. The Characteristic-Locus (CL) method is based on the extension of the eigenvalues and the eigenvectors of a square matrix of constants for multivariable control system design.

2.1.2 Decentralized (Multi-loop) control

In decentralized or multi-loop control, MIMO systems are controlled using multiple Single Input Single Output (SISO) controllers or MIMO controllers. Decentralized control is usually applied, if the process is composed of different time domains, which makes it possible to use several control loops without a need of a multi-variable controller. Moreover, decentralized control is preferred for cases where active constraints remain constant, in contrast to cases with changing active constraints, where a multi-variable controller is applied. The difficulties, arising from the use of decentralized control are the interactions of a single input to multiple outputs. The more interactions between the multiple loops arise, the harder is it to control the system.

In the present work, only the concept of decentralized control is considered. In fact, in the further progress of the work the decentralized control is referred as multi-loop control, to underline the interactions, caused by the multi-loop control structure. Moreover, the focus is set on SISO controllers. In the further control system design only PI controllers are considered. Usually, PI controllers are preferred to PID controllers in certain types of processes for simplicity, when dealing with non-intensive variables, or in systems with a significant noise. In practice, the derivative term of the PID controller could amplify disturbanced inputs or noise in the case that the PID is not well tuned. If ramps or other kinds of time functions are used as references, extensions to the use of PID controllers could be made.

The use of PI and PID controllers, in the following abbreviated with PI(D), in multi-loop control systems is known in the literature and preferred in the industry due to a simple implementation and the possibility of manual tuning. In the following section, the PI(D) control in multi-loop control systems is introduced and the arising problem as a result of loop interactions is demonstrated with an example.

2.1.3 PI(D)-Control in Multi-loop Control systems

The mutual interactions of control loops in multi-loop control systems produce a considerable problem in control system design, compared to SISO systems. The interactions are able to destabilize a system. Opening or closing one loop could result in a change of the dynamics in the other control loops. The following example will demonstrate the problem ((Johnson u. Moradi, 2005)).

Consider a system with two inputs u_1 and u_2 and two outputs y_1 and y_2. The corresponding dynamics is given by

$$y_1 = \frac{1}{s+1}u_1 + \frac{\gamma}{s+1}u_2 \tag{2.1}$$

$$y_2 = \frac{\delta}{s+1}u_1 + \frac{1}{s+1}u_2 \tag{2.2}$$

The control structure of the process is displayed in Fig. 2.1, with interaction terms, linked to the parameters γ and δ.

The process set points are denoted as r_1 and r_2 for the outputs y_1 and y_2, respectively. Assumed, input u_1 controls output y_1 using a PI-controller with parameter k_1 given by

$$u_1 = k_1(1 + \frac{1}{s})(r_1 - y_1) \tag{2.3}$$

and input u_2 is set to zero. Then the closed-loop transfer function is

$$\frac{y_1}{r_1} = \frac{k_1}{s+k_1}. \tag{2.4}$$

Equation (2.4) is a stable transfer function for any $k_1 > 0$. In the same way, if the system is controlled such that input u_2 controls output y_2 using a PI-controller, where it's parameter k_2 is given by

$$u_2 = k_2(1 + \frac{1}{s})(r_2 - y_2) \tag{2.5}$$

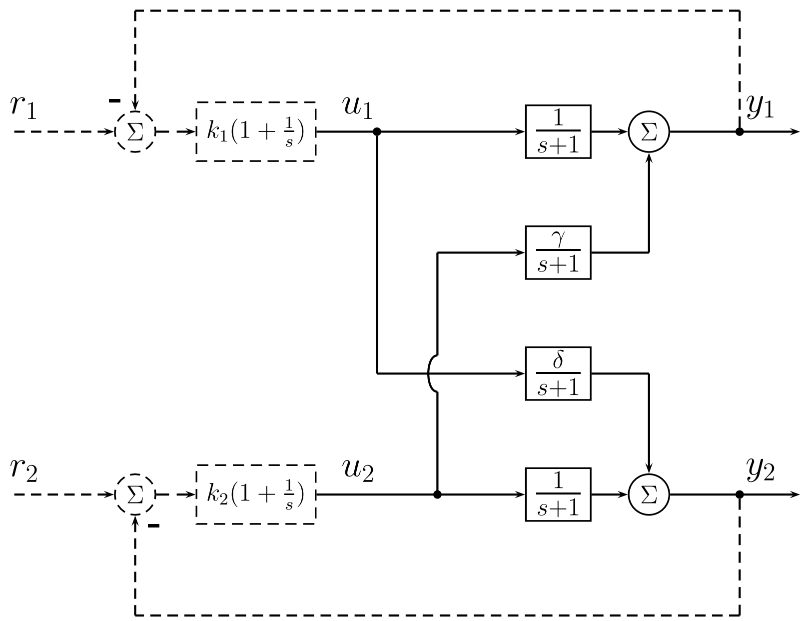

Figure 2.1. Interactive process of a MIMO system with two inputs u_1, u_2, two outputs y_1, y_2 and two possible control loops.

and the first input u_1 is set to zero. Then the closed-loop transfer function is

$$\frac{y_2}{r_2} = \frac{k_2}{s + k_2}. \tag{2.6}$$

Similarly, equation (2.6) is stable whenever $k_2 > 0$. But, when both loops are closed and both PI-controllers operate together, the transfer function from r_1 to y_1 is

$$\frac{y_1}{r_1} = \frac{s + 1 - \gamma\delta}{s^2 + 2s + 1 - \gamma\delta} \tag{2.7}$$

with $k_1 = 1$ and $k_2 = 1$. Notice, equation (2.7) is unstable if $\gamma\delta > 1$. This example shows, that the interaction can destabilize a system of individually stabilized single-loops in a multi-loop structure. This motivates the research in this area.

As pointed out in Chapter 1, only few methods for control system design in multi-loop systems are known in the literature. The methods can be classified into three subgroups, dependent on how the loop interactions are considered during the control system design:

(I) The control loop interactions are completely neglected ((Johnson u. Moradi, 2005), (Qiang Xiong u. He)).

Method: The controller parameters of each control loop are tuned apart. Afterward, all parameters are retuned with a common factor to stabilize the total system and to obtain an adequate load disturbance rejection.

(IIa) The control loop interactions are partly considered (Lunze, 2004).
Method: In decoupling the loop interactions are considered insofar as a mathematical decoupling of the different control loops is done first and then tune the controller parameters of the each loop. With the decoupling, the interactions are compensated mathematically. Using this method, other control theoretic problems may occur.

(IIb) The control loop interactions are partly considered (only dominant loop interactions) ((Bristol, 1966), (Rosenbrock, 1974)).
Other multivariable control system design techniques, like the RGA or the Gershgorin circles measure the control loop interactions and find the best pairing of controlled and manipulated variables. Nyquist techniques try to make the transfer function diagonal dominant using a compensator matrix. The focus of these methods is set on dominant control loop interactions, if they exist.

(III) The control loop interactions are completely considered during the parameter tuning.
A method where the control loop interaction are completely considered including the consideration of several system requirements is missing. Independent design methods exists, that tune the controller parameters on paired transfer functions and considering some constraints due to process interactions (Qiang Xiong u. He).

Thus, the request for a control system design method for multi-loop systems, considering loop interactions, exists.
Game theory contributes an essential part to the new approach as it is used as a modelling tool, which is specified in more detail in the following.

2.2 Introduction to game theory

2.2.1 History and spheres of influence

The domain of game theory is not new in research. It's roots lie in works of von Neumann in the years of 1928−1937 and Borel in the years of around 1920 (Luce u. Raiffa, 1989). At the beginning, there was not much interest. Due to the fact, that the original writings were written by and for mathematicians. Researchers with less mathematical background loose their motivation to deal with the reasoning and conclusions of game theory. For this reason, the published works of this time rest for around two decades. Only the last world war refreshed the interest on game theory due to growing demand for military strategies, which was an important factor for it's further rapid development.
Since game theory is not new in the field of research, the literature delivers several different definitions of a game. Summarizing the statements of a few authors, the core area of game theory could be stated as:

- Game theory analyses situations in which (Holler u. Illing, 2000)
  - (i) The final outcome depends on the decisions of several players.
  - (ii) Every player knows about this strategic interdependence.
  - (iii) Ever player assumes that all the other players equally know about this strategic interdependence.
  - (iv) Every player considers (*i*), (*ii*) and (*iii*) in his or her's [1] decisions.
- A situation of decision-making, where multiple players track their individual goals and make decisions to reach their goals (Riecks, 2006).
- Instrument to analyse strategic situations of decision-making (Turocy u. v. Stengel, 2001).
- Game theory is a collection of mathematical models to study situations of conflict and/or cooperation (Lemaire, 1991).

[1] The player is said to be male in all further considerations although player could also be female

- Game theory is a mathematical theory of rational strategy selection used to analyse optimal choices of two or more actors or players. Each player has preferences for all possible outcomes (Brams, 1990).

Summarizing, game theory can be described as a tool, modelling situations, where more than one player makes decisions which mutually interact and whose objectives are in conflict.
A general and important assumption in game theory is the property of the players rationality. A rational player will play according to the rules and each player will act optimally depending on their goals. Beyond that it is essential that every player knows that the other players are rational and that every player knows that the other players know that they are rational as well and so on. The field of Game theory can be divided in several ways and several layers, depending on the properties of a game. For example, there are distinctions in strategic (static) games and dynamic games, in cooperative and non-cooperative games, and in games of complete or incomplete information, to name only a few.
To get more insight, the most common rules, dominated in the literature and defined by famous game theorists, are summarized and specified in the following. The majority of introductions to game theory start with a definition of one of these games: (Basar u. Olsder, 1999), (Holler u. Illing, 2000), (Luce u. Raiffa, 1989), (Riecks, 2006) or (Osborne u. Rubinstein, 2001). But a more general description of a game is given in (J. Neumann, 2004):

"A game is a totality of rules which describes it."

Hence, according to (Vincent u. Leitmann, 1970), the rules instruct the players how to play the game. That means, the rules prescribe each player's cost function, they prescribe the system manipulated by the players and set the limitations on the player's control strategies. The rules also contain information about the properties of the game. All details that are important in the further work will be specified in the following subsection.

2.2.2 Common definitions, rules and properties of a game

2.2.2.1 Number of players N

The number of players N varies from game to game, but must already be defined before the game is played. Basic two-player games are addressed in detail in the literature. If possible, an extension to n-player games can be performed, too.

2.2.2.2 Strategies u

Concerning the strategies u_i with $i = 1, ..., N$ there is a big disagreement in the literature. For instance, (Ferguson) act on the assumption that in cooperative games the strategies should be neglected. Since the main features in cooperative or coalition games are those of a coalition (that means: who joined the coalition), and the value of the coalition. Others, like (McCain) ask the question:

> What strategy choice will lead to the best outcome for all players in the game?

A further question may be:

> How large a bribe may each player reasonably expect for choosing it.

However, the definition of a strategy independent of a cooperative or a non-cooperative game, that should be valid in this work, can be formulated as:

> A nonempty set $\mathcal{U}_i$ is called the action space of player i. Each $u_i \in \mathcal{U}_i$ is referred to as a strategy (LaValle, 2006).

A further distinction concerning the strategies can be drawn in deterministic (pure) and stochastic (mixed or randomized) strategies. With pure strategies, a player chooses a strategy with probability 1. While with mixed strategies, the decision of the player is given through a probability distribution of the available strategies.

2.2.2.3 Payoffs, Costs and Outcomes

The payoffs in a game are often called outcomes or costs. In this work, the payoffs as well as the corresponding payoff functions are called costs and cost functions, respectively, because of their implication of minimization required in the later work.

In a many player game, each player i has a cost function J_{cf_i}, which he seeks to minimize (Vincent u. Leitmann, 1970). The index cf specifies the type of cost function. The cost functions J_{cf_i} are defined on $\mathcal{U}_1 \times \ldots \times \mathcal{U}_i \rightarrow \mathbb{R}^+ \cup \infty$. The set of possible cost pairs the players can obtain is called the utility set. Remark, it is not naturally that pairs specifies a pair only of two. Setting up the cost functions is one of the main and often one of the most difficult parts in describing a game. The fashion in which the players employ their control choices toward that end depends on the mood of play.

2.2.2.4 Dynamic vs. static game

According to (Basar u. Olsder, 1999), a dynamic game is defined to be a game in a dynamic decision process, that evolves a (discrete or continuous) time period with more than one decision maker. Each decision maker possess his own cost function with different access to different information. In contrast to dynamic games, (Basar u. Olsder, 1999) defines a static game as a game, where the players act only once and independent from each other. There is only one round of decision making, thus it is called a one shot game.
The author of (Isaacs, 1999), who is also a pioneer in the field of game theory, defines a static and dynamic game as follows:

> In conventional game theory (static games) a strategy of a player consist of a decision set, that tells him, which move he should play for every possible game position, appearing during the game. If the player chooses their strategies, the outcomes of the game are completely determined.

In contrast, a game with a variable decision process is called a dynamic game. The choice of the control variable is a function of the state variables. For every possible game position that can occur, there exist a set of values for the state variables. Each player chooses a set of values during the decision making. These represent their control variables.
Remark, a dynamic game can be distinguished again between a differential and a difference game. Usually, in the literature, a dynamic game and a differential

game are set equal, but this is not regular. The differences of a differential and a difference game will be specified in this section as well.

2.2.2.5 Cooperative vs. non-cooperative games

The literature delivers several different definitions between cooperative and non-cooperative games ((Vincent u. Leitmann, 1970),(Basar u. Olsder, 1999)). According to (Vincent u. Leitmann, 1970), two or more players cooperate while playing a game if they help each other to minimize their respective cost functions, and as long as they do not degrade them.
In (Basar u. Olsder, 1999), the cooperative game theory is not addressed. The reasons are, that cooperative games can, in general, be reduced to optimal control problems by determining a single cost function to be optimized by all the players. This property would suppress the game aspects of the problem.
This statement is not wrong, but dealing with one single cost function involves some disadvantages for the players. Using a single cost function, usually the weighted sum method is applied. The weighted sum method requires a weighting of the different aspects, depending on their importance. The assignment of the weights is still a problem as well as the disadvantage that the weighted sum method is not able to take care of all conflicting design objectives individually (Bernard, 2005). The weighted sum method belongs to scalarization methods which do not always give satisfying solutions, because of interest conflicts of design objectives and possible compensations within the cost functions.

2.2.2.6 Pure bargaining games vs. transferable utility games

According to (Hart u. Mas-Colell, 1997), usually two special classes of games are distinguished in the field of cooperative games: pure bargaining games and transferable utility games.
In pure bargaining games, only the grand coalition matters with the value for the coalition S : $\mathcal{S} = \left\{ fp \in \mathbb{R}^{\mathcal{S}} \text{ such that } fp_i \leq 0 \; \forall \; i \in \mathcal{S} \right\} \;\; \forall \; S \neq N$. $\mathcal{S}$ is the set of feasible outcomes for the coalition S. A bargaining game is described through a number of players N and a pair $(\mathcal{P}, d)$, where $\mathcal{P}$ represents the feasible set and d gives the disagreement point of the game and consists of the costs, if the players

do not negotiate (Holler u. Illing, 2000). Thus, if cooperation fails, the players end up at the disagreement point d. The characteristics of a bargaining game are, that at least one cost vector $\mathbf{J} = (J_{cf_1}, ..., J_{cf_N})$ lies in $\mathcal{P}$, providing smaller costs than d. A solution problem arises only if more than one cost vector J is smaller than d. The problem of selecting a particular point in the utility set $\mathcal{P}$ is called the bargaining problem, usually axiomatic bargaining, as well.
In contrast, in transferable utility games, also known as games with side payments, only one single number represents what a coalition can get, and this amount is arbitrarily divided among the members of the coalition.

2.2.2.7 Differential vs. difference games

Both type of games belong to the dynamic games, whose dynamic decision process evolve over time. The difference is given through the time domain which is either discrete or continuous, respectively it is about a difference game or a differential game.

2.2.2.8 Normal form games vs. extensive form games

In a normal form representation of a game, every player i chooses a strategy u_i, without knowledge of the decisions of the other players. A normal form game, which is also known as matrix game or strategic game, has no dynamic and is strictly non-cooperative, since no coalitions are considered. Using the extensive form game, the sequential course of a game is represented. It specifies at which point of time which player is acting, which strategies are available for the acting player and the knowledge the acting player possesses. If all players choose their strategies for the complete game progress at the beginning, the normal form is an appropriate description for the decision making process. The advantage of the normal form game representation is the comparatively simple way for the determination of Nash solutions through the method of static optimization (Osborne u. Rubinstein, 2001). When transferring from the extensive form to the normal form a loss of information is obtained.

2.2.2.9 Zero-Sum vs. Nonzero-Sum Game

A game is called a zero-sum game, if players have opposite objectives. This means what one player wins (J), the other player will loose ($-J$). Therefore the sum of the different cost functions is always zero $\sum_{i=1}^{N} J_{cf_i} = 0$. In contrast, nonzero-sum games always have a value which is different from 0 with $\sum_{i=1}^{N} J_{cf_i} \neq 0$.

2.2.2.10 Information

The information that underlies in a game is an essential part in game theory. Basically, the literature distinguishes between the following different information:

(I) Perfect vs. imperfect information
In a game with perfect information, every preliminary actions of the other players are known (also known as perfect recall). Each player knows at each time in which information set he (or she) is. The information set contains only one information node, which gives information about all possible moves (chosen strategies) that have taken place in the game. In contrast, in a game with imperfect information, several information nodes in the information sets exist and the players often do not know in which information node they are. Thich means, they do not know which strategies the other players have choosen.

(II) Complete vs. incomplete information
In a game with complete information, every player knows the strategy sets $\mathcal{U}_i$ and the costs J_{cf_i} of all players at each time and there are no private information. Such a game is easy to analyze and is often called boring. Only a game with incomplete information is said to be interesting, since, for example, the cards the players hold in their hands are not visible for each player. Thus, they have private information.

Note, a game with complete information could also be a game with imperfect information. For example, all strategy sets $\mathcal{U}_i$ and costs J_{cf_i} of all players are given, in spite of the existence of several elements of information sets in the game.

Which information type is valid for a game is given through the information structure of a game. It specifies perfect or imperfect information as well as complete or incomplete information.

2.2.2.11 Game of Kind vs. Game of degree

According to (Isaacs, 1999) the quantity in a game of degree is the payoff (cost value), whereas the criteria in a game of kind are discrete. Typical examples of games of kind are pursuit evasion games, possessing only two outcomes: either the evader is caught or not.

The definitions, rules and properties specified in the current subsection are limited on the essential and used as the general basis for the further development of the game-theoretic approach. Only the concepts needed for the development of the approach are given. This is required, since game theory is a wide research field and not only limited to control theory. For instance, game theory is also a research field in finance, economics, assertion, law and others. According to this wide area of application, many academic literature is published, defining many different games, rules, solution concepts and their applications with sometimes opposed definitions.

2.2.3 Solution Concepts in Game Theory: An Overview

A solution concept in game theory associates a set of feasible costs for each game. Ideas to design solution concepts for non-cooperative games are based on the equilibrium concept, while the solution concepts for cooperative games are based on keywords like justice, equity, fairness and stability.

2.2.3.1 Solution concepts for non-cooperative games

Non-cooperative games are commonly described as matrices or trees in the two or three player case. Especially for a two-player game, the subsequent solution concepts are directly applicable to matrix games, where also a graphical implementation of the solution concepts is possible. However, all presented solution concepts are applicable for n-player games as well.

(I) Normal (strategic) form game solutions (according to (J. Neumann, 2004))

(a) Dominant strategy:
A control strategy $u^* = (u_1^*, \ldots, u_m^*)$ with property $J_{cf_i}(u_i^*, u_{-i}) \leq J_{cf_i}(u_i, u_{-i}) \ \forall \ u = (u_i, u_{-i}) \in J_{cf_i}$ for all u_i^* is called an equilibrium in dominant strategies. A distinction is made between strictly dominant strategies with $J_{cf_i}(u_{i*}, u_{-i}) < J_{cf_i}(u_i, u_{-i})$ and weakly dominant strategies with $J_{cf_i}(u_{i*}, u_{-i}) \leq J_{cf_i}(u_i, u_{-i})$. Every player i chooses his strategy u_i, independent of the other players' behavior.

(b) Maximin solution:
A control strategy $u^* = (u_1^*, \ldots, u_m^*)$ is called a minimax solution, if $J_{cf_1}(u_1^*, u_2^*) \leq J_{cf_1}(u_1, u_2^*)$ and $J_{cf_2}(u_1^*, u_2^*) \geq J_{cf_2}(u_1^*, u_2)$. The minimax solution is a saddle point and it is often used in non-cooperative games with more than one Nash equilibrium, but it is not always applicable, since there is no guarantee for the existence of a saddle point solution.

(c) Nash equilibrium:
A control strategy $u^* = (u_1^*, \ldots, u_m^*)$ is called a Nash equilibrium, if $J_{cf_i}(u_i^*, u_{-i}^*) \leq J_{cf_i}(u_i, u_{-i}^*)$. No coalition is assumed, so each player is acting independently. The equilibrium solution is secure against any attempt by one player unilaterally to alter his strategy. Also, it is assumed that every player is using his Nash control, if a given player plays non-Nash-optimally, he will do no better, and similar for every other player. Summarized, there is no incentive for any player to deviate from his strategy.

(d) Correlated strategies (Osborne u. Rubinstein, 2001):
Every mixed strategy Nash equilibrium corresponds to a correlated equilibrium. Using mixed strategies, probabilities are assigned to each strategy, a pure strategy for example is assigned with probability 1.

(e) Rationalizable strategies (Osborne u. Rubinstein, 2001):
Every strategy used with positive probability in a correlated equilibrium is rationalizable. A strategy is a never-best response if and only if it is strictly dominated. Strategies that survive iterated eliminations of strictly dominated strategies are rationalizable.

(II) Extensive form game solutions

(a) Nash equilibrium (Osborne u. Rubinstein, 2001):
In determining the Nash equilibrium for an extensive game, the sequential structure of the game is ignored. The strategies are treated as choices that are made once and for all before play begins, compare item Id).

b) Sub game perfection:
A strategy set is sub game perfect, if the solution is an equilibrium for all possible sub games.

Remark, all solution concepts for normal form games, specified in item (I) could be applied to extensive form games and the other way around. Since both representations are mutually transformable.

2.2.3.2 Solution concepts for cooperative games

The solution concepts for cooperative games needs partially more mathematical effort compared to the concepts for non-cooperative games. However, the possibility for a graphical solution representation is provided for two-player games as well.

(III) Pure bargaining games (or individualist games)

(a) Nash bargaining solution (NB) ((Ehtamo u. Hämäläinen), (Luce u. Raiffa, 1989)):
The players find the Nash bargaining solution $N(\mathcal{S})$ simply by maximizing Nash's product

$$N(\mathcal{S}) = max(u_1 - d_1) \cdot (u_2 - d_2) \tag{2.8}$$

in $\mathcal{S}$ with d as disagreement point. According to the notion of John Nash, the Nash bargaining solution includes a fair negotiation resolution, accepted from the rational players. The function ϕ, defined through (2.8), assigns to each bargaining game $(\mathcal{P}, d)$, exactly one strategy vector $\mathbf{u} = \{u_{1_j}, \ldots, u_{N_j}\}$, the Nash solution, and satisfies the following four axioms:

(1) Scale invariance:
For every bargaining game $(\mathcal{P}, d)$ and for arbitrary real numbers $m_i > 0$ and w_i, with $i = 1, 2$, $\phi_i(\mathcal{P}'d') = m_i\phi_i(\mathcal{P}, d) + w_i$, if $(\mathcal{P}', d')$ is a bargaining game, resulting from a linear transformation, keeping the order of all elements u and d in $\mathcal{P}$, so that $y_i = m_i x_i + w_i$ and $d' = m_i d_i + w_i$ and y and d' elements of $\mathcal{P}'$.
In words, N(S) is independent of the units, so the solution does not vary if the utility is multiplied by a positive constant.

(2) Symmetry:
If $\mathcal{P}, d$ is a symmetric bargaining game, then $\phi_1(\mathcal{P}, d) = \phi_2(\mathcal{P}, d)$.

(3) Independence of irrelevant alternatives:
$f(\mathcal{P}, d) = \epsilon(\mathcal{Q}, d)$ if $(\mathcal{P}, d)$ and $(\mathcal{Q}, d)$ are bargaining games with equal disagreement point d, $\mathcal{P}$ is a subset of $\mathcal{Q}$ and $f(\mathcal{Q}, d)$ is an element of $\mathcal{P}$.

(4) Pareto optimality:
$(\mathcal{P}, d)$ is a bargaining game; if $x_1 \leq \phi_1(\mathcal{P}, d)$ and $x_2 \leq \phi_2(\mathcal{P}, d)$, then $x \neq f(\mathcal{P}, d)$ in $\mathcal{P}$.

(b) Egalitarian solution (E) (Holler u. Illing, 2000):
The egalitarian solution is a special case of the proportional solution (PR). The basic idea of the PR solution is as follows: at a crossover from one bargaining game $(\mathcal{P}, d)$ with $\mathcal{P}$ for the feasible set and d for the disagreement point, to another bargaining game $(\mathcal{R}, d)$ with equal disagreement point d and an arbitrary size of utility set, and $\mathcal{P}$ is a subset of $\mathcal{R}$, all players should get equal payoff increases, being proportionally related. The egalitarian solution provides an exact uniform distribution of the gains resulting from the cooperation.

(c) Kalai-Smorodinsky solution (KS):
The best known variation of the Nash bargaining solution is the Kalai-Smorodinski solution. Kalai and Smorodinsky replaced the third Nash axiom for the Nash bargaining solution by the monotonicity axiom:

(3') Monotonicity.
If the negotiation set (feasible set) $\mathcal{P}$ is enlarged such that the minimum utilities of the players remain unchanged, then neither of the players must not suffer from it.

According to (3′), the Kalai-Smorodinsky solution is situated at the intersection of the Pareto-optimal curve and the straight line linking the disagreement point and the utopia point UP (Holler u. Illing, 2000).

(IV) Transferable utility games (or coalitional games):

(a) Core ((Holler u. Illing, 2000), (Osborne u. Rubinstein, 2001)):
The idea of the core is to look at those payoff vectors which no coalition can improve upon. The core collects costs J_{cf_i} (also called imputations) that are not dominated. All possible payoff pairs are imputations where none of the players gets less than he would get if he plays alone. In general the core is a selection from the set of imputations. For two player games the set of imputations coincides with the core.

(b) Banzhaf index (Dubey u. Shapley, 1979):
Using the Banzhaf index as solution method, the number of coalition when an agent is pivotal out of all winning coalitions containing that agent is counted. It is used for measuring *real power* in weighted voting systems, whereas *power* is defined as: which agent has the most influence on the outcome.

(c) Shapley value:
The Shapley value is a solution concept, that assigns the average of marginal contributions to coalitions. A single payoff for each player is described, which is the average of all marginal contributions of that player to each coalition he is a member of.

A graphical interpretation of the solution concepts for bargaining games is given in Fig. 2.2 with the Nash bargaining solution (NS), the Egalitarian solution (E), the Kalai-Smorodinski solution (KS) and the Proportional solution (PR) for a free chosen example.

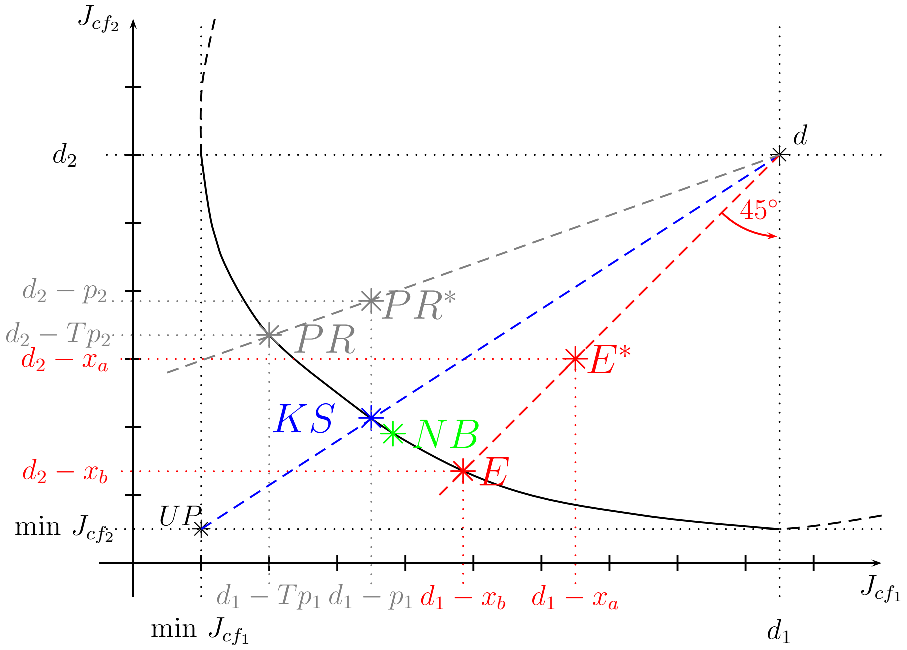

Figure 2.2. Solution concepts for bargaining games. Black: proportional solution, red: Egalitarian solution, blue: Kalai-Smorodinsky solution, green: Nash bargaining solution.

The Pareto-optimal set acts as a part in obtaining a solution in cooperative bargaining games. Pareto-optimality is a measure of efficiency. The outcome of a game is Pareto-optimal if it cannot be improved upon without hurting at least one player. Pareto-optimality offers a set of efficient solutions that are more or less preferable from a players viewpoint. A Pareto-optimal strategy describes a socially optimal joint strategy for cooperative games. A Pareto-optimal point is defined in (Marler u. Arora, 2004) as follows:

> A point, $\alpha^* \in A$, is Pareto-optimal iff there does not exists another point, $\alpha \in A$, such that $J(\alpha) \leq J(\alpha^*)$, and $J_{cf_i}(\alpha) < J_{cf_i}(\alpha^*)$ for at least one function.

Remark, an equilibrium in dominant strategies (compare Ia)) is not Pareto-optimal since individual rationality is faced with collective rationality.
A few disadvantages considering Pareto-optimality are scheduled in (Farina u. Amato). The fact that the number of improved or equal objectives is not taken into account belongs to it, as well as the (normalized) size of improvements is not taken into account. Another aspect is the not considered preference among the objectives, but this is a property of the a-priori methods solving multi-objective optimization problems.
The comparison of the weighted sum solution and the Nash bargaining solution on an example in the two dimensional space is given in Fig. 2.3, where the weights of the weighted sum method are chosen as 0.5 each. According to Fig. 2.3, the

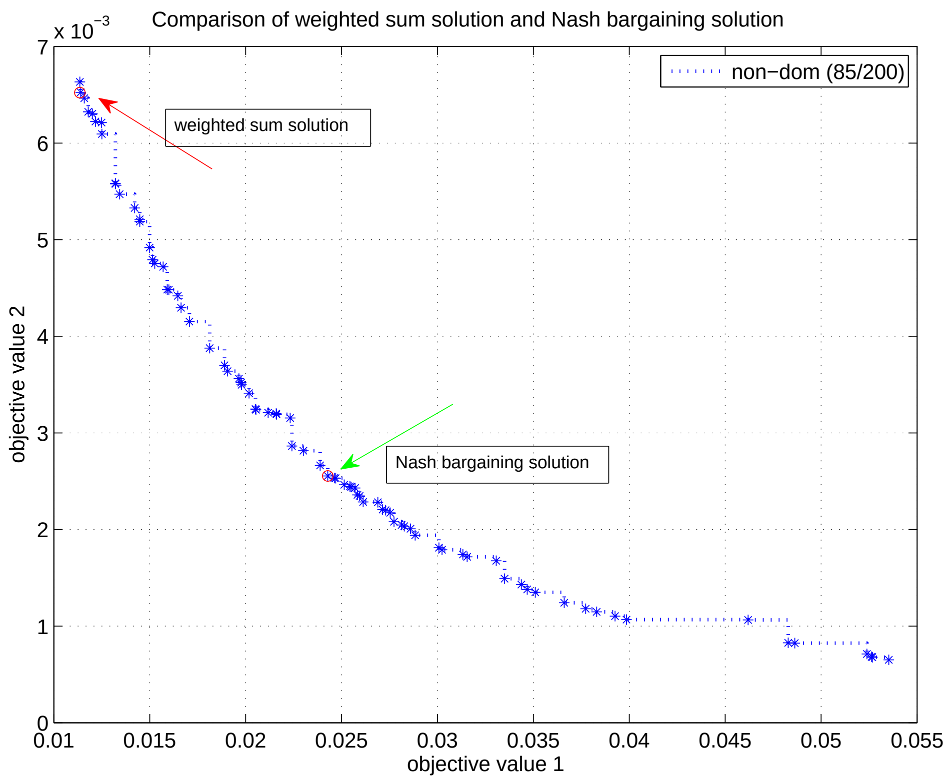

Figure 2.3. Pareto-optimal front in the case of two cost functions. Comparison of the weighted sum solution with weights $\beta_1 = \beta_2 = 0.5$ and the Nash bargaining solution.

weighted sum shows the minimum cost function value in J_1 (objective value 1) with respect to the cost function distribution for the range of J_2 (objective value

2).In contrast, the Nash bargaining solution provides a solution which is fair and equal distributed, considering both cost function ranges for both cost functions. The final solution of the bargaining game is mostly obtained through a specified operation with the Pareto-optimal set, see 2.2.3.2. The final solution could not be obtained until the Pareto-optimal set is available. Thus, it is illustrated how the Pareto-optimal set is derived using multi-objective optimization.

2.3 Multi-objective optimization (MOO)

2.3.1 Multi-objective optimization providing a Pareto-optimal solution set

The goal of multi-objective optimization is to find a vector $\boldsymbol{\alpha}$ of decision variables or parameters, which satisfies constraints and optimize more than one cost function J_{cf_i}. In the present work, cost functions J_{cf_i} mathematically describe the cost criteria, which are usually in conflict with each other. In (de Weck, 2004), methods for multi-objective optimization are distinguished in scalarization methods and Pareto methods. Scalarization methods merge different cost functions J_{cf_i} in one general cost function J, Pareto methods in contrast, keep the different cost functions J_{cf_i} and optimize them in common.
The usual way in scalarization methods is to accumulate the different objectives J_{cf_i} to one J by using the weighted sum method with the weights, β_i (Andersson, 2000; Bernard, 2005).

$$J_{sum} = \sum_{i=1}^{N} \beta_i J_{cf_i} \tag{2.9}$$

As mentioned in Subsection 2.2.2, the scalarization methods not always give satisfying solutions because of interest conflicts of design objectives. Multi-objective optimization using Pareto methods is, on the contrary, able to take care of all conflicting design objectives individually but compromising them concurrently (Bernard, 2005). The key concept of multi-objective optimization is the Pareto-optimality. However, the set of Pareto-optimal solutions is usually combined with high computational effort. This computational effort is based on the set of all possible solutions that has to be calculated, all representing the Pareto-optimal set. According to

(Saksala, 2004), a cooperative game is formalized as a multi-objective optimization problem. Due to the fact that one is dealing here with several individual players, which should be treated equally, the characteristics of Pareto methods is the most appropriated approach for cooperative dynamic games. Next, the mathematical problem of multi-objective optimization is formalized.

2.3.1.1 Multi-objective mathematical optimization problem

A multi-criteria optimization problem for the mathematical programming is formulated in (Osyczka, 1985) as follows:
Find a vector $\boldsymbol{\alpha}^*$ such that

$$J(\boldsymbol{\alpha}^*) = opt J(\boldsymbol{\alpha}) \tag{2.10}$$

and such that it will satisfy υ inequality constraints

$$\eta_\iota(\boldsymbol{\alpha}) > 0 \ for \ \iota = 1, 2, ..., \upsilon \tag{2.11}$$

and ξ equality constraints

$$h_l(\boldsymbol{\alpha}) = 0 \ for \ l = 1, 2, ..., \xi < \psi \tag{2.12}$$

where

1. $\boldsymbol{\alpha} = [\alpha_1, \alpha_2, ..., \alpha_\psi]^T$ is a vector of decision variables defined in ψ-dimensional Euclidean space of variables $\mathcal{E}^\psi$,

2. $J(\boldsymbol{\alpha}) = [J_{cf_1}(\boldsymbol{\alpha}), ..., J_{cf_i}(\boldsymbol{\alpha}), ..., J_{cf_N}(\boldsymbol{\alpha})]$ is a vector function defined in N-dimensional Euclidean space of objectives $\mathcal{E}^N$,

3. $\eta_j(\boldsymbol{\alpha}), h_l(\boldsymbol{\alpha})$ and $J_{cf_i}(\boldsymbol{\alpha})$ are linear and/or nonlinear functions of decision variables $\boldsymbol{\alpha}_1, ..., \boldsymbol{\alpha}_\psi$.

The constraints given by $\eta_i(\boldsymbol{\alpha})$ and $h_l(\boldsymbol{\alpha})$ represent the restrictions imposed to the optimization problem.
In multi-criteria optimization problems the task is either to:

1. minimize all the cost functions

2. maximize all the cost functions

3. minimize some and maximize others

However, a cost function that has to be maximized is converted to a cost function that has to be minimized as follows:

$$\max_{i} \ J_{cf_i}(\boldsymbol{\alpha}) = \min_{i} \left(-J_{cf_i}(\boldsymbol{\alpha})\right). \tag{2.13}$$

The solution of multi-objective problems is a set of points known as the Pareto-optimal set. The optimum in the sense of Pareto gives a set of also called non inferior solutions. Non inferior solutions are solutions for which there is no way of improving any cost of objective without leading to a degradation of at least one other. The set of feasible points fp such that there exists $\boldsymbol{\alpha} \in \mathbb{R}^N$ where $fp = J(\boldsymbol{\alpha})$ is called the *attainable set* or *feasible set*, denoted by $\mathcal{P}$:

$$\mathcal{P} = \left\{fp|\ \exists\ \boldsymbol{\alpha} \in \mathbb{R}^N :\ fp = J(\boldsymbol{\alpha})\right\} \tag{2.14}$$

Existence conditions for the *attainable set* were developed by (Clarke u. Gawthrop, 1997) and (Dutta u. Vetrivel, 2001) for convex multi-objective optimization problems.
The *utopia point* (UP) is defined as the point in the utility space with coordinates given by the solutions of the scalar optimization problems:

$$\min J_{cf_i} \ , i \in \{1, \ldots, N\} \tag{2.15}$$

A graphical representation of the main definitions in multi-objective optimization problems is presented in Fig. 2.4 for two cost functions J_{cf_1} and J_{cf_2}, which have to be minimized. $J^*_{cf_1}$ and $J^*_{cf_2}$ represent the minima of the respective cost function.

2.3.2 Solving Multi-objective optimization problems

In the field of multi-objective optimization, generally three methods are distinguished:

- No-preference method:

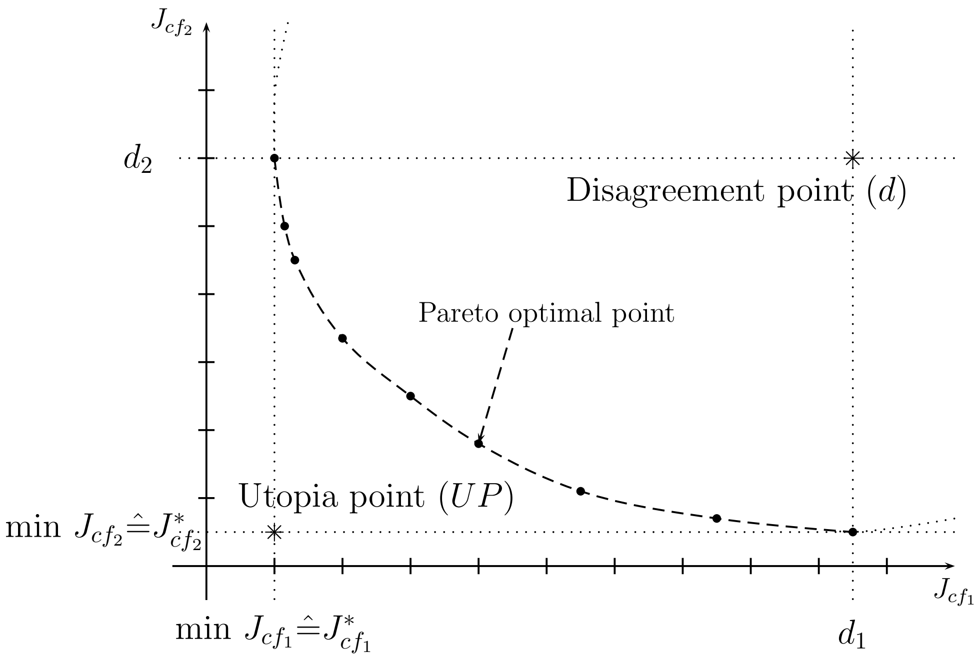

Figure 2.4. Graphical representation of the main definitions in MOO. Disagreement point d, utopia point UP and Pareto-optimal front as dashed line.

In using a no-preference method, one final solution is obtained. The algorithm results in one final solution and not a complete Pareto-optimal front without previous specification of preferences. No decision maker is required. An example of a no-preference method is the method of the global criterion, where the solution point is chosen, according to a special metric, next to the utopia point, but concurrently is part of the reachable solution set. Disadvantages of the no-preference method lie in the final single solution. For example, if a satisfying solution is not reached, there is no possibility to engage the optimization. Also, there is no choice among several possible solution points, which would give some kind of robustness to the solution.

- A-priori method:
  The decision maker of an a-priori method specifies the preferences and objectives in advance. For instance, the function is arranged according to the importance, to affect the final solution prior to the starting of the algorithm.

Again, the final solution is a single point and no Pareto set, with the advantage of less time consumption for the calculation. The time expensive computation of many Pareto-optimal points is left out. An example for an a-priori method is the weighted sum method, where the multiple objectives are weighted according to their importance and summed up for the optimization process. Among the disadvantages from one final solution of the no-preference method, the a-priori method has the difficulty that the result depends only on the choice of the preferences. According to (Makowski, 1994), there is typically no general way to aggregate all criteria into one objective that can adequately represent a preference structure of a decision maker.

- A-posteriori method:
  The decision maker of an a-posteriori method specifies no preference or valuation of the objective function in advance. The task of the decision maker is the selection of one final solution out of the Pareto-optimal set, so it is applied after the method provides a set of Pareto-optimal solutions. Known examples for a-posteriori methods, providing Pareto-optimal sets, are the normal boundary intersection algorithm (NBI), the multi-objective genetic algorithm (MOGA) or the non-dominated sorting genetic algorithm II (NSGA-II). The calculation effort increases with the number of solutions.

The choice, which method to use, strongly depends on the type of the problem.
Since the final solution of the bargaining game is basically determined through the Pareto-optimal set, in being part an a-posteriori method is chosen for the future work.
A common method, resulting in a Pareto-optimal set, is to use genetic algorithms. The advantages of using a genetic algorithm to solve a MOO problem compared to other solution methods are: 1) GA's are Pareto methods, which are able to take care of all conflicting design objectives individually but compromising them concurrently (Bernard, 2005), 2) GA's have a multiple search property (Kawabe u. Tagami, 1999) and 3) convex as well as nonconvex Pareto-optimal fronts could be obtained (Konak u. a., 2006).
Disadvantages of using GA's are given through a large amount of settings that

influence it's computation time and the resolution of the Pareto-optimal set. The computation time states how many computation steps are needed to get a satisfying Pareto-optimal solution set. The resolution of the Pareto-optimal set is important later on, when selecting a final solution.
Next, a short introduction to genetic algorithms will be given.

2.3.2.1 Genetic algorithms for the solution of MOO problems

Genetic algorithms are adaptive methods regarding search and optimization problems. First steps and works with genetic algorithms are made by (Holland, 1992) in the years of around 1990.
Genetic algorithms follow the process of natural behavior. Starting point is a population of individuals, whereas each of them represents a possible solution to the given problem. The fitness values of each individual are mutually compared, while each fitness value is calculated using a performance function,. The fitness value gives an index, how close it is to the target value. Those individuals, relating to satisfying fitness values are especially appropriate to reproduce themselves in the next generation. Those who produce a nonsatisfying fitness value do not reproduce themselves and die out. Resultant in a new and better population of possible solutions and extending several generations, the good properties are passed on. If the genetic algorithm has been designed well, the population converges to an optimal solution (Beasley u. a., 1993) if only one cost function is considered.
The efficiency of a genetic algorithm, i.e. how *good* is a solution, evaluated according to computation time and resolution, mainly depends on the choice of the representation of the variable format, the calculation of the fitness value, the selection method, the recombination method, the mutation method and the reinsertion method. The genetic algorithm, applied to obtain a Pareto-optimal set is available as toolbox for Matlab. For the present work, the used variable format representation is chosen as real valued, the calculation of the fitness value is described in detail in (Pohlheim, 2001). As selection method, stochastic universal sampling is used, where the individuals are mapped to contiguous segments of a line such that each individual's segment is equal in size to its fitness. Then, equally spaced pointers are placed over the line as many as there are individuals to be selected. Discrete recombination is applied as recombination method which is more specified

in (Pohlheim, 2001). The mutation of the variables is also real valued, whereas randomly created values are added to the variables with a low probability. The reinsertion occurs locally, that means the individuals are selected in their bounded neighborhood. It is possible, that other setting combinations of the genetic algorithm yield better results but is not yet improved in this work.
One advantage feature of genetic algorithms is the robust and parallel search technique, which is applied in many areas. Although it is not guaranteed that the global optimal solution set could be found if it even exists, satisfactory solutions can be found in a relative short amount of time. Thus, genetic algorithms are a qualified method that enables parallel computing for a relative fast formation of the Pareto-optimal set.

2.3.3 The Genetic Evolutionary Algorithm Toolbox

The first version of the Genetic Evolutionary Algorithm Toolbox (GEATbx) for Matlab was developed in 1995 by Hartmut Pohlheim. Since 1995, the toolbox was enhanced continuously. Of course, there are other toolboxes for solving MOO problems, like NSGA-II or NBI, but just this toolbox was chosen, since it is a toolbox for use with Matlab. The documentation is clear and satisfactory, a related book is released, it was directly applicable even for multi-objective optimization, it provides a Pareto-optimal set in the multi-objective case and Mr. Pohlheim was available anytime for questions, concerning the toolbox. A comparison of other available software, solving multi-objective optimization problems, is proposed in (Natto, 2007).

Chapter 3

Game-theoretic control system design

The presented theoretic preliminaries of Chapter 2 provide the basis for the game-theoretic approach of the multi-loop control system design, elaborated in this chapter. The approach is composed of:

(I) A game description of a multi-loop control system.

(II) Cost functions and constraints set up, considering different system requirements.

(III) A Pareto-optimal set as solution set of the game, obtained through MOO, which is solved using a GA.

(IV) A solution concept for the final solution, chosen from the Pareto-optimal set.

In the main part of the present thesis, the developed approach is limited to transfer function models that do not include any delays. Additionally, the control system design is kept very simple and does not consider elements for anti-windup or output range. The ambition for the solution of the game is to obtain a trade-off within all, often conflicting, cost functions. The details of the approach are given in this chapter.

3.1 Description of the game

The multi-loop control system design is specified as a dynamic game, which is described either in continuous time or in discrete time, resulting in a differential game or a discrete game, respectively. The different controllers in the multi-loop control structure represent the players of the game. Each player has to satisfy at least one cost function J_{cf_i} using the available strategies u_{i_j} of each players' strategy set. With cf indexing the kind of cost function, $i \in \{1, \ldots, N\}$ indexing the player and $j \in \{t, k\}$ indexing the different strategies of player i at time t or k in the continuous and differential case, respectively. The cost functions are a formalization of the posed requirements on the system as well as on the single control loops, as for example a low control effort, or robust stability, or a good disturbance reaction, see Section 1.1. The cost functions often depend on each other because of the multi-loop control structure, and thus are in conflict in meeting the requirements.

3.1.1 The Continuous Game

A dynamic game, passing in the continuous time domain is called a differential game. In general, a dynamic differential game is a system with the following properties according to (Lygeros u. a., 1997) with appropriate upgrading:

(a) The game is described on the time period $[t_0, T]$.

(b) The game consists of N players (persons, controllers, ...) with $N > 1$.

(c) Each player i with $i = 1, \ldots, N$ dispose of the control variable u_{i_j} with $u_{i_j} \in \mathcal{U}_i$, the input space or strategy space; the elements of $\mathcal{U}_i$ are denoted as $\left\{u_{i_j} \hat{=} u_i(t), t_0 \leq t \leq T\right\}$ and are allowed controls of player i.

(d) The differential game is described through differential equations $x_i^{(n)}(t)$:

$$x_i^{(n)}(t) = f_i(x_i^{(n-1)}(t), \ldots, \dot{x}_i(t), w_1(t), \ldots, w_m(t)) \tag{3.1}$$

where the general solution contains n arbitrary variables which correspond to n constants of integration, with w_j, where $j = 1, \ldots, M$, representing the inputs (what is given) and x_i representing the outputs of the game (not comparable to the outcome of the game).

(e) Each player i possesses his own cost function J_{cf_i} defined on

$$J_{cf_i} : X_i \times \mathcal{W}_1 \times \ldots \times \mathcal{W}_M \rightarrow \mathbb{R}^+, \tag{3.2}$$

which he tries to optimize. Additionally, it is possible, that a player possesses more than one cost function, which the player tries to optimize.

(f) Each player i disposes his own game strategy (game concept, control law) $u_{i_j} \in \mathcal{I}_i$, which determines the control (strategy) u_{i_j} from the information set $\mathcal{I}_i$.

(g) Each player i possesses an information set $\mathcal{I}_i$, which is mainly composed of

   (a) Differential equation of the system $\dot{x}_i(t)$.

   (b) General solution for x_i.

   (c) Own cost function J_{cf_i}.

   (d) Own game strategy $u_{i_j} \in \mathcal{U}_i$.

   (e) Game strategy of other players $u_{\neg i_j}$.

The information structure of the game is either perfect or imperfect with the difference being complete or incomplete.

3.1.2 Multi-loop control system design of a differential game

For the description of the differential game, it is assumed, that the plant is modelled by the coprime rational expression

$$\frac{Y_{\#p}(s)}{U_{\#p}(s)} = G_{\#p}(s) = \frac{B_{\#p}(s)}{A_{\#p}(s)} \tag{3.3}$$

with $\#p = 1, \ldots, \#P$ as number of processes and

$$\frac{B_{\#p}(s)}{A_{\#p}(s)} = \frac{b_{n_B} s^{n_B} + b_{n_B-1} s^{n_B-1} + \ldots + b_1 s + b_0}{s^{n_A} + a_{n_A-1} s^{n-1} + \ldots + a_1 s + a_0} \tag{3.4}$$

The control law is given by

$$U_i(s) = C_i(s)E_i(s) = \frac{Q_i(s)}{P_i(s)}E_i(s) \tag{3.5}$$

with i indexing the multiple control loops of the system, which is equivalent to the number of players.
The polynomial description of the PI controllers C_i with proportional parameters K_{P_i} and integral parameters K_{T_i} is

$$C_i = \frac{Q_i}{P_i} = \frac{K_{P_i}s + K_{P_i}/K_{T_i}}{s}. \tag{3.6}$$

To realize the differential game description of Subsection 3.1.1, the inputs and outputs of the game in subchapter 3.1.1 (d) have to be specified, see Fig. 3.1. In the multi-loop control system design the input of the corresponding game (w_j in subchapter 3.1.1 (d)) consists of the plant model polynomials $A_{\#p}(s)$ and $B_{\#p}(s)$, the controller polynomials Q_i and P_i, as well as the reference variables $r_i(s) \in R_i(s)$.
These polynomials enable the derivation of one particular solution from the general solution of 3.1.1 (d) by setting these constants to particular values. The output of the game are the control signals U_i (x_i from 3.1.1 (d)).

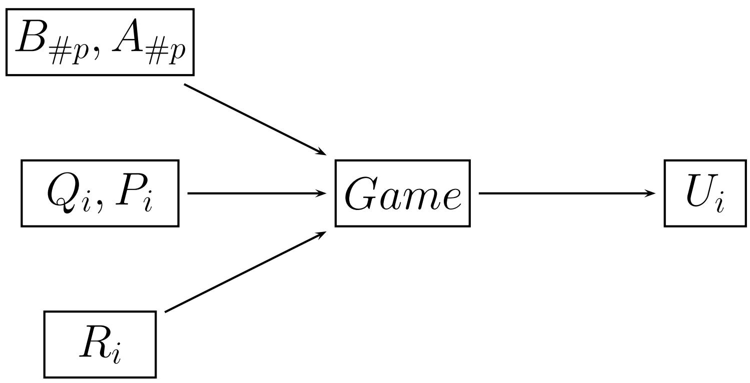

Figure 3.1. Inputs and outputs of a game

This results in an equation of the form

$$x_i^{(n)} + \ldots + l_1\dot{x}_i + l_0 x_i = o_m w_j^{(m)} + \ldots + o_1 w_j + o_0 \tag{3.7}$$

The transfer function relating the outputs x_i to the inputs w_j represent the error that exist in the multi-loop control system. The number of error equations $e_i(t)$ depends on the number of control loops, from the game-theoretic view. It is the number of players, N.

The game can now be described as a differential game between N players with $i = 1, \ldots, N$ on the time period $[t_0, T]$. The strategies of the players are defined as

$$u_i(t) = \int_{t_0}^{T} c_i(t)e_i(t-\tau)d\tau \tag{3.8}$$

with

$$\mathcal{L}\{c_i(t)\} = C_i(s) = Q_i(s)/P_i(s). \tag{3.9}$$

Q_i and P_i are the controller parameters of player (controller) C_i. The strategies of the players belong to the strategy sets $\mathcal{U}_i = \left\{u_{i_j} | u_{i_j} \textit{ is given by } (3.8)\right\}$.

The differential game can now be described as the evolution of the errors e_i with

$$e_i^{(n)}(t) = f(e_i^{(n-1)}(t), \ldots, \dot{e}_i(t), u_1(t), \ldots, u_N(t)) \tag{3.10}$$

and initial condition

$$e_i(t_0) = e_{i0} \tag{3.11}$$

as well as cost functions J_{cf_i} with

$$J_{cf_i} = g_{i0}(e_{iT}). \tag{3.12}$$

The errors e_i belong to the set $E_i = \{e_i | e_i \textit{ as solution of} (3.10)\}$. Function f_i is defined on $f_i : R_1 \times \ldots \times R_I \times \mathcal{U}_1 \times \ldots \times \mathcal{U}_N \rightarrow \mathbb{R}^+$ and function g_{i0} on $g_{i0} : R_1 \times \ldots \times R_I \times \mathcal{U}_1 \times \ldots \times \mathcal{U}_N \rightarrow \mathbb{R}^+$. with R_I indexing the reference value(s).

The terminal state e_{iT} as well as the cost functions J_{cf_i} depend on the choice of $u_{1_j}, \ldots, u_{N_j}$. Important for the proposed controller tuning method is the dependence of the players strategies $u_{1_j}, \ldots, u_{N_j}$ on the controller parameters Q_i and P_i, as well as the control laws of the system and the reference signals r_{0i}.

3.1.3 The difference game

It is advantageous, because controllers are normally implemented in digital computer systems, to describe the game in the time-discrete case, as well.
The dynamic differential game of Subsection 3.1.1 is transferred to a dynamic difference game with the following properties:

(a) The game is described on the discrete time period $[k_0, K]$.

(b) The game consists of N players (persons, controllers, ...) with $N > 1$.

(c) Each player i with $i = 1, \ldots, N$ dispose of the control variable u_{i_j} with $u_{i_j} \in$ input space or strategy space $\mathcal{U}_i$; the elements of $\mathcal{U}_i$ are denoted as $\left\{u_{i_j} \hat{=} u_i(k), k_0 \leq k \leq K\right\}$ and are allowed controls of player i.

(d) The difference game is described through difference equations $x_i(k+n)$:

$$x_i(k+n) = f_i(x_i(k+n-1), \ldots, x_i(k+1), w_1(k), \ldots, w_M(k)) \tag{3.13}$$

where the general solution contains n arbitrary variables which correspond to n constants of integration. With w_j, $j = 1, \ldots, M$, representing the inputs (what is given) and x_i representing the outputs of the game (not comparable to the outcome of the game).

(f) Each player i possesses his cost function J_{cf_i}

$$J_{cf_i} : X_i \times \mathcal{W}_1 \times \ldots \times \mathcal{W}_M \rightarrow \mathbb{R}^+, \tag{3.14}$$

which he tries to optimize. Again, it is possible, that a player possesses more than one cost function J_{cf_i}.

(g) Each player i dispose an own game strategy (game concept, control law) $u_{i_j} \in \mathcal{I}_i$, which determines the control strategy u_{i_j} from the information set $\mathcal{I}_i$.

(h) Each player i possesses an information set $\mathcal{I}_i$, which is mainly composed of

   (a) Differential equation of the system $x_i(k+n)$.
   (b) General solution for $x_i(k)$.
   (c) Own cost functions J_{cf_i}.
   (d) Own game strategy $u_i \in \mathcal{U}_i$.
   (e) Game strategy of other players $u_{\neg i_j}$.

   Again, the information structure of the game is either perfect or imperfect, as well as complete or incomplete.

3.1.4 Multi-loop control system design as a difference game

For the description of the difference game, it is assumed, that the plant is modelled by the coprime rational expression

$$\frac{Y_{\#p}(z)}{U_{\#p}(z)} = G_{\#p}(z) = \frac{B_{\#p}(z)}{A_{\#p}(z)} \tag{3.15}$$

with $\#p = 1, \ldots, \#P$ as number of processes and

$$\frac{B_{\#p}(z)}{A_{\#p}(z)} = \frac{b_{n_B} z^{n_B} + b_{n_B-1} z^{n_B-1} + \ldots + b_0}{z^{n_A} + a_{n_A-1} z^{n_A-1} + \ldots + a_0}. \tag{3.16}$$

The control law is given by

$$U_i(z) = C_i(z) E_i(z) = \frac{Q_i(z)}{P_i(z)} E_i(z) \tag{3.17}$$

with i indexing the multiple control loops of the system, respectively the players of the game.
The polynomial description of the PI controllers C_i with proportional parameters

K_{P_i} and integral parameters K_{T_i} is

$$C_i = \frac{Q_i}{P_i} = \frac{K_{P_i} z + K_{P_i}/K_{T_i}}{z-1}. \tag{3.18}$$

To realize the difference game description of Subsection 3.1.3, the inputs and outputs of the game in subchapter 3.1.3 (d) have to be specified, see Fig. 3.1. In the multi-loop control system design the input of the corresponding game (wj in subchapter 3.1.3 (d)) consists of the plant model polynomials $A_{\#p}(z)$ and $B_{\#p}(z)$, the controller polynomials Q_i and P_i, as well as the reference variables $r_i(z) \in R_i(z)$.
These polynomials enable the derivation of one particular solution from the general solution of 3.1.3 (d) by setting these constants to particular values. The output of the game are the control signals U_i (x_i from 3.1.3 (d)).
This results in an equation of the form

$$x_i(k+n)+\ldots+l_1x_i(k+1)+l_0x_i(k) = o_m w_j(k+m-1)+\ldots+o_1w_j(k)+o_0 \tag{3.19}$$

The transfer function relating the outputs x_i to the inputs w_j represent the error that exist in the multi-loop control system. The number of error equations $e_i(k)$ depends on the number of control loops, in the game-theoretic view it is the number of players, N.
The game can now be described as a difference game between N players with $i = 1, \ldots, N$ on the time period $[k_0, K]$. The strategies of the players are defined as

$$u_i(k) = c_i(k) * e_i(k) \tag{3.20}$$

with $i = 1, ..., N$ and $\mathcal{Z}\left\{c_i(k)\right\} = C_i(z) = Q_i(z)/P_i(z)$.
The controller parameters of C_i are contained in Q_i and P_i. The strategies of the players are part of the strategy sets $\mathcal{U}_i = \{u_i | u_i \text{ is given by (3.20)}\}$.
The difference game is described as the evolution of the errors e_i with

$$e_i(k+n) = f_i(e_i(k+n-1), \ldots, e_i(k), u_1(k), ..., u_N(k)) \tag{3.21}$$

and initial condition $e_i(k_0) = e_{i0}$ as well as a cost function J_{cf_i} with

$$J_{cf_i} = g_{i0}(e_{iK}). \tag{3.22}$$

The errors e_i are part of the set $E_i = \{e_i | e_i \text{ as solution of (3.21)}\}$. The function f_i is defined on $f_i : R_1 \times \ldots \times R_I \times \mathcal{U}_1 \times \ldots \times \mathcal{U}_N \rightarrow \mathbb{R}^+$ and function g_{i0} is defined on $g_{i0} : R_1 \times \ldots \times R_I \times \mathcal{U}_1 \times \ldots \times \mathcal{U}_N \rightarrow \mathbb{R}^+$ witn I indexing the reference value(s).

The final state e_{iT}, as well as the cost functions $J_{i_{ec}}$, depend on the choice of $u_i(k)$. Again, the strategies $u_i(k)$ of the players depend on the controller parameter Q_i and P_i, as well as on the control structure and the reference signals r_{0i}.

3.2 Cost functions and constraints set up

3.2.1 Reference tracking

The basic requirement on the control loops of the multi-loop system should be set on a good reference tracking. The stability is transferred to a constraint while the requirements on a good reference tracking for each control loop are converted to cost functions.

A typical performance index, applied to control problems, achieving a good reference tracking, is the Integral Square Error (ISE) with

$$J_{ISE_i} = \int_{t=0}^{\infty} e_i^2(t)dt. \tag{3.23}$$

for the continuous case, and

$$J_{ISE_i} = \sum_{k=0}^{\infty} e_i^2(k). \tag{3.24}$$

for the discrete case.

A second common performance index, applied to control problems for a fast reference tracking with low deviation, is the Integral of Time weighted Square Error

(ITSE). Including the consideration of the elapsed time k with

$$J_{ITSE_i} = \sum_{k=0}^{\infty} k e_i^2(k). \quad (3.25)$$

for the discrete case.
The third performance index, considered in this work, is the Integral of Square Time weighted Square Error (ISTSE)

$$J_{ISTSE_i} = \sum_{k=0}^{\infty} k^2 e_i^2(k). \quad (3.26)$$

for the discrete case.
All three performance indices J_{cf_i} with $cf \in \{ISE,\ ITSE,\ ISTSE\}$ are applied for the discrete case and compared, as their use will lead to different Pareto-optimal sets. Equations (3.23) and (3.24) are solved according to (Aström, 1970) during the course of the game. Equations (3.25) and (3.26) are solved according to (Gambier, 2007).
Satisfying a fast reference tracking with low deviation of a system is a basic requirement of the initial game description. To conform to more requirements, the game is extended in the following to requirements considering the control effort, control constraints, robustness and stochastic disturbances. Those requirements are mathematically formalized as additional cost functions and/or constraints.

3.2.2 Control effort

According to the fact that, in practice, every control signal cannot be followed by the physical system. The proposed method should provide the opportunity either to minimize the control effort or even to limit the control effort of the plant. All possibilities are established.

3.2.2.1 Control effort as add on to existing cost functions

A first approach is proposed in (Wellenreuther u. a., 2006b) with the cost functions J_{cf_i} use the ISE (Integral Square Error). Some authors (Isermann, 1989) include a constraint for the control signal to the performance signal. As control signals do not necessarily converge to zero which is important for J_i to be $\leq \infty$, the square of the control signals derivative of $u_i(t)$, or difference of $u_i(k)$ is used instead. In (Wellenreuther u. a., 2006a), the square of the control signal's difference $\Delta u_i(k)$ is integrated, added using a weighting factor to the discrete version of the ISE and successfully applied on an example. Some authors, like (Isermann, 1989) or (Kawabe u. Tagami, 1999), already included in the performance index as a constraint for the control signal. The derivative square of the control signal is multiplied with a weighting factor λ_i and added to the squared error signal $e_i(t)$, namely

$$J_{eu_i} = \int\limits_{t=0}^{\infty} e_i^2(t)dt + \lambda_i \int\limits_{t=0}^{\infty} \dot{u_i}^2(t)dt \tag{3.27}$$

for the continuous case, and

$$J_{eu_i} = \sum_{k=0}^{\infty} e_i^2(k) + \lambda_i \sum_{k=0}^{\infty} \Delta {u_i}^2(k), \tag{3.28}$$

$$J_{eu_i} = \sum_{k=0}^{\infty} k e_i^2(k) + \lambda_i \sum_{k=0}^{\infty} k \Delta {u_i}^2(k), \tag{3.29}$$

$$J_{eu_i} = \sum_{k=0}^{\infty} k^2 e_i^2(k) + \lambda_i \sum_{k=0}^{\infty} k^2 \Delta {u_i}^2(k) \tag{3.30}$$

for the discrete case.

The cost functions J_{eu_i} in (3.27) - (3.30) are actually some sort of a *weighted* sum involving two main disadvantages during the optimization process. The first one is the possible compensation of the squared error signal $e_i(t)$ and the squared derivative of the control signal $\dot{u_i}(t)$, while optimizing their sum. The second disadvantage is the choice of the weights λ_i. If λ_i is set to 1, i.e. the derivative of the control signal is completely incorporated during the optimization process, the constraints on the control signals may be violated.

3.2.2.2 Control effort as cost function implementation

The second implementation, concerning the requirement on a low control effort, is to consider the square of the control signals derivative separately, using a cost function:

$$J_{u_i} = \int_{t=0}^{\infty} \dot{u_i}^2(t)dt \tag{3.31}$$

for the ISE implementation continuous case, and

$$J_{u_i} = \sum_{k=0}^{\infty} \Delta u_i^2(k) \tag{3.32}$$

for the ISE implementation of the discrete case. The advantage is that the cost function for the control effort is treated equally during the optimization process in the control system design. The disadvantage is constraints on the control effort can not be set explicitly. The ITSE and ISTSE implementations could be easily extracted from the last part of (3.29) and (3.30), respectively.

3.2.2.3 Explicit control constraints implementation

To be able to keep limits for control signals according to predefined step changes of the systems' set points, explicit constraints are added to the game in (Wellenreuther u. a., 2007). The constraints are maintained during the offline optimization of the cost functions. From the game-theoretic view, constraints on the control signal u_{i_j} equals constraints on the strategy sets $\mathcal{U}_i$ of the players.

If the control signals u_{i_j} of the plant, where u_{i_j} represents the continuous case $u_i(t)$ as well as the discrete case $u_i(k)$, are limited, constraints are set around an operating point in a predefined range of $\pm L_i$:

$$|u_i| \leq L_i. \tag{3.33}$$

These constraints are added to the dynamic game with the present challenge to optimize the cost functions (3.23)-(3.26), concerning the error convergence subject to (3.33).

Note, these constraints are set for the offline optimization of the cost functions and there is no guarantee that these limits are kept online.

3.2.2.4 Explicit control constraints as cost function implementation

Additionally, the idea is to keep the deviation of the control signals to the linearization point as small as possible. To obtain the smallest possible control signal deviation, the distance between the control signal $u_i(t)$ and the limits $\pm L_i$ should be as large as possible. Additionally, these constraints can be reformulated as

$$max(L_i - |u_i|). \tag{3.34}$$

Constraints in MOO problems are treated in the literature in several ways. A very popular way is to add penalty functions to the cost functions. In (Pohlheim, 2000), the penalty function is modelled as a weighted sum. By assigning the values of the weights, the importance of the compliance with the constraints is defined. Again, the well known handicap in the use of the weighted sum persists: the determination of the weights. Another way to handle constraints is to check them during the calculation of the cost functions. If constraints are violated, the corresponding cost function values are ignored in the optimization process. A survey of constraint handling techniques in evolutionary computation methods is given in (Michalewicz, 1995).
Further, it is possible to formulate the constraints as cost functions. Since every cost function is minimized in the game, (3.34) can be rewritten as cost function for the control signals J_{cu_i} with explicit constraints as

$$J_{cu_i} = -(L_i - |u_i(t)|). \tag{3.35}$$

3.2.3 Robust Stability

The difficulty of describing a physical process as a mathematical model is specified in (Skogestad u. Postlethwaite, 1996), and (Manoso u. a., 1997) as the robustness problem. The robustness problem is solved first by characterizing the uncertainty and incorporating it into the mathematical model. In the literature, uncertainty is distinguished between two main classes: parametric uncertainty and uncertainty caused by unmodelled dynamics, (Balas u. a.), (Skogestad u. Postlethwaite, 1996).

In the case of parametric uncertainty, the structure of the model, including the order, is known, but some parameters are uncertain. This type of uncertainty can be modelled as inverse additive uncertainty, (Becerra). In contrast, unmodelled dynamics occur due to the high frequency plant behavior, which is often uncertain since only the low order nominal model describing the low-mid frequency range behavior of the plant is available. One common approach to model this type of uncertainty is to use a multiplicative uncertainty model, (Skogestad u. Postlethwaite, 1996).

The singular value analysis σ, which is a generalization of the Nyquist criterion, is becoming popular as a general way to analyse the robust stability of MIMO systems.

The structured singular value μ of a transfer function matrix M, where M represents a known linear system, is defined as $\mu(M) = 1/\sigma(M)$ subject to the singular value. It was developed to analyse the effects of parametric uncertainties and unmodelled dynamics to the stability and the performance of multi-loop control systems. The structured singular value μ is defined on finding the smallest structured perturbation Δ, measured in terms of $\sigma(\Delta)$, which makes $\det(I - M\Delta) = 0$, with I as unity matrix, then $\mu(M) = 1/\sigma(\Delta)$.

The peak of the frequency response of the general structured singular value μ delivers, depending on the structure of the perturbation, the size for the perturbation where the closed loop system remains stable. A value of $\mu = 1$ represents a perturbation with $\sigma(\Delta) = 1$. If smaller perturbations makes the system unstable, the value of μ is larger than 1 and if the value of μ is smaller than 1, larger perturbations are permitted.

A robust stability theorem for block-diagonal perturbations is given in (Skogestad u. Postlethwaite, 1996):

> Assume that the nominal system M and the perturbations Δ are stable. Then the $M\Delta$-system is stable for all allowed perturbations with $\bar{\sigma}(\Delta) \leq 1$, $\forall\omega$, if and only if $\mu(M(j\omega)) < 1$, $\forall\omega$.

Considering the requirement on robust stability during the control system design, a cost function J_μ for the total system is defined as

$$J_\mu = \mu(M). \tag{3.36}$$

Concerning the robust stability requirement, only the cost function (3.36) is optimized by all participant players. The value of the robust stability cost J_μ that has to be optimized, depends on the players' control strategies. Considering the cost J_μ of (3.36) with regard to the solution of the game, an additional trade-off between the robust stability and the performance of the system subject to constraints on the control strategies has to be met.

Finally, for all cost functions, it is imperative that a threshold on the cost functions could be set, to distinguish from unacceptable performance to acceptable performance, where the control wins the game against the environment or not.

3.3 Course of the game

The game passes during the simultaneous optimization of the cost functions J_{cf_i}. The cost functions are calculated, depending on different strategy combinations $\{u_1(t),\ \ldots,\ u_N(t)\}$ or $\{u_1(k),\ \ldots,\ u_N(k)\}$, and compared in incorporating their evaluation over the total time intervals $[t_0, \ldots, T]$, and $[k_0, \ldots, K]$, respectively.
Using a genetic algorithm for the calculation of the Pareto-optimal set, the calculations of the cost functions are done in parallel to a certain degree. The number of parallel calculations is specified through the number of subpopulations as well as the population size. Based on a starting population given through a set of randomly chosen parameters for the polynomials Q_i and P_i within a specified range, those strategies (parameters of the polynomials, leading to the control signals u_i (strategies)) *survive*, leading to a minimization of the cost functions while keeping the system requirements. The surviving strategies are given through points on the Pareto-optimal set.
If the cost functions are in conflict, as is usually the case, trade-offs are made within the cost functions. Using the *survivals* as the result of one generation, algorithms for selection, recombination, mutation and reinsertion, provided by the genetic algorithm, are applied on the *survivals*. Obtained parameter sets for Q_i are used to calculate the cost functions for the new generation and the new *survivals* are saved. After a predefined number of generations, the algorithm stops and the obtained generation provides the Pareto-optimal set with the largest number of non dominated points, that is the Pareto-optimal set with the highest resolution.

3.4 Solution of the game

According to (J. Neumann, 2004), the solution of a cooperative game is a set of solutions. All non dominated solutions, also called Pareto-optimal solutions, are part of this Pareto-optimal set.
In the further progress of this work, the Pareto-optimal set is identified first for the application examples. For the determination of a final solution, out of the Pareto-optimal set, a decision maker (DM) is required.

3.4.1 Motivation for a game-theoretic DM

All points on the Pareto-optimal set deliver non-dominated solutions in the utility set, representing trade-offs within the predefined control-theoretic requirements of the control system, converted into cost functions, constraints and including the consideration of the control loop interactions. That is, in the utility set of two cost functions: if the worth of the first cost function is improved (for example, the integrated squared error of the belonging control loop is decreased), the worth of the second is degraded (for example, the integrated squared error of the belonging control loop is increased).
Obtaining a Pareto-optimal set, from which the final solution is chosen, implies that no priorities are set in this work a-priori. If so, the weighted sum approach as a-priori method could be used using only one cost function and resulting after the optimization in only one final solution. Also, the priorities could be set using a lexicographic ordering and optimization method, providing only one final solution. Both methods make no use of a DM.
In contrast, one possible implementation of a DM is to define a further crucial control-theoretic requirement. The final solution is then obtained in checking all Pareto-optimal points on the additional requirement and choosing the point, satisfying the specified requirement the most, which is known as an a-posteriori method. For example the Pareto-optimal set, obtained through the minimization of the errors of two control loops, implemented with two cost functions, is now

checked on robust stability, or the final solution should give the Pareto-optimal point with the smallest overshoot for the control signal of the second loop. Many other decision makers, motivated from control theory, are applicable. The proposed game-theoretic approach could be modified, by changing part (IV) of the presented method: the final solution concept. In doing so, it depends on a particular system and specific requirements, which is not in the sense of the present work.

In contrast, when using the game-theoretic approach, it is assumed, that all control theoretic requirements of the system are formulated at the beginning and treated equally. This equitable usage of the requirements leads to the demand on the DM to be fair to all requirements, as well. In the research field of game theory, the problem of choosing one single point of the Pareto-optimal set is known as the bargaining problem or bargaining game.

According to (Hart u. Mas-Colell, 1997), usually two special classes of games are distinguished in the field of cooperative games: pure bargaining games and transferable utility games. In pure bargaining games, only the grand coalition matters. The grand coalition is the coalition where all game participants (players) unite. A two person bargaining game without transferable payoffs (utilities) is often called a two person bargaining game ((Lemaire, 1991)). In any bargaining game, a solution should satisfy Pareto-optimality, because it guarantees that there exists no other outcome, preferred by each player in agreement. Two common solution concepts for Nash bargaining games, compare (Holler u. Illing, 2000), (Ehtamo u. Hämäläinen), and (Luce u. Raiffa, 1989), could be applied as decision maker to choose a final solution from the Pareto-optimal set for control theoretic applications. The two solution concepts are the Nash bargaining solution and the Kalai-Smorodinski solution, which are described in detail in Subsection 2.2.3.2. All solution concepts are presented for comprehension in the two dimensional case in Appendix B, where the Nash bargaining solution emerge as final solution concept, applied in this approach.

3.5 Essential modifications in the source code of the genetic algorithm

To obtain the Pareto-optimal set with the highest resolution, changes in the available GA had to be performed. In the original version of the GA of (Pohlheim, 2001), there is no specified method in selecting the final solution out of the Pareto-optimal set. One abort criterion of the algorithm is the maximum number of generations, which is used in this work. The original version collects all Pareto-optimal points of each generation with the ordering top fitness value first. Then, a comparison of the top cost functions of each generation is implemented, while only the values of the first cost function are compared. This is only a temporary solution of the genetic algorithm and therefore modifications in the code of the genetic algorithms are necessary.

Hence, the genetic algorithm is modified in collecting all Pareto-optimal points of each generation. The generation with the highest number of Pareto-optimal points is chosen as final Pareto-optimal set as it is the set with the highest resolution. After obtaining the Pareto-optimal set a final solution concept could be applied.

Chapter 4

Game-theoretical Topological Analysis of a Two-input/Two-output System

In the field of game theory, information is an essential component. Hence, the game-theoretic view of controller design in multi-loop control systems allows for the question, how unequally distributed or incomplete information affects the selection of the players' strategies. Different information sets lead to different strategy selections provided by the controllers and the belonging control laws. To study and discuss the effects of different information sets to the solution of each game, an asymmetric triangular multi-loop control structure is chosen as basis. The derivation of the error equations for the different topologies, developed in this chapter, is given in Appendix C.

4.1 Game description

The basic control structure for the multi-loop system is shown in Fig. 4.1. The two-input/two-output system consists of two controllers $C_{11} = Q_{11}/P_{11}$, and $C_{22} = Q_{22}/P_{22}$, and three transfer functions $G_{11} = B_{11}/A_{11}$, $G_{21} = B_{21}/A_{21}$, and $G_{22} = B_{22}/A_{22}$ describing the process through polynomial equations.
The given control structure is triangular (asymmetric) in such a way that the upper control loop act as a disturbance on the lower control loop. Thus, the control

loops of the multi-loop system interact only in one-way. The control system design with optimal performance concerning a reference tracking with minimum error convergence is now implemented as a cooperative differential game.

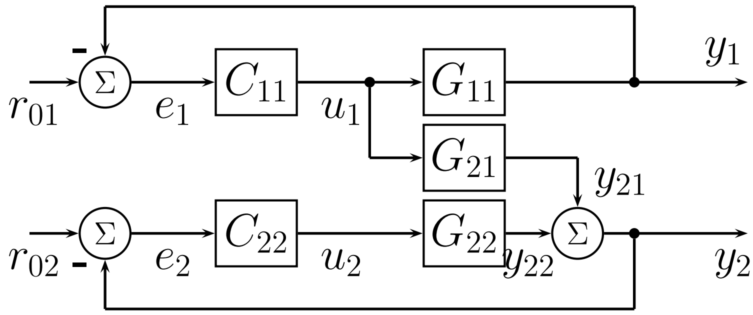

Figure 4.1. Triangular control structure of a TITO system.

4.1.1 The cooperative differential game

The control system design of the two-input/two-output system in Fig. 4.1 is considered as a differential game between two players i with $i = 1, 2$ on the time period $[t_0, T]$. The strategies of the players are defined as

$$u_i(t) = \int_{t_0}^{T} c_{ij}(t) e_i(t - \tau) d\tau \tag{4.1}$$

with

$$\mathcal{L}\{c_{ij}(t)\} = C_{ij}(s) = Q_{ij}(s)/P_{ij}(s). \tag{4.2}$$

Q_{ij} and P_{ij} with $j = 1, 2$ are the controller parameters of C_{ij} in Fig. 4.1. The strategies of the players belong to the strategy sets $\mathcal{U}_i = \{u_i | u_i \textit{ is given by } (4.1)\}$. The differential game can now be described as the evolution of the errors e_i with

$$e_i^{(n)} = f(e_i^{(n-1)}, \ldots, \dot{e}_i, u_1, u_2) \tag{4.3}$$

and initial condition

$$e_i(t_0) = e_{i0} \tag{4.4}$$

as well as a cost J_{cf_i} with

$$J_{cf_i} = g_{i0}(e_{iT}). \tag{4.5}$$

The errors e_i belong to the set $E_i = \{e_i | e_i \ as\ solution\ of (4.3)\}$. Function f_1 is defined on $f : R_1 \times \mathcal{U}_1 \to \mathbb{R}$, function f_2 is defined on $f : R_1 \times R_2 \times \mathcal{U}_1 \times \mathcal{U}_2 \to \mathbb{R}$ and function g_{i0} on $g_{i0} : R_i \times \mathcal{U}_1 \times \mathcal{U}_2 \to \mathbb{R}$.

4.1.2 Information

An essential component in the field of game theory is the information. According to (J. Neumann, 2004), a player can make decisions (choose strategies) only dependent on his available information at that time. In this chapter, the available information is given through the control system structure.

To be able to classify the present information structure in a game, game theory distinguishes - among others - between complete and incomplete information, see Chapter 2. Summarizing the main facts of Subchapter 2.2.2, in a game with complete information, all players know the strategy sets $\mathcal{U}_i$ and the costs J_{cf_i} at any time. There are no private information like unknown strategies u_i for player $\neg i$ or even unknown payoffs J_{cf_i}. In contrast, in a game with incomplete information, certain properties as for example the controller parameters of a player i are unknown to the team mates. A further distinction is done with symmetric and asymmetric incomplete information, where all players do not know a parameter, and asymmetric incomplete information, where only some of the team mates do not know this parameter.

Many games are characterized through unequally distributed or incomplete information. This is exactly the case which is given in the triangular control structure of Fig. 4.1. In this structure, player 1, that is the controller of the upper control loop C_{11} which operates with his own information, is the control law of the upper control loop and the parameters of controller C_{11}. In contrast, player 2, that is the controller of the lower control loop C_{22} has information about his own control law and the parameters of controller C_{22} as well as information about player 1, indicated by the information flow from player 1's control signal u_1 over the transfer function of G_{21} to the output signal y_{22} of player 2.

In the following, different possible information structures and thus different control system design games with different control structures are implemented to be able to study their effects on the control system behavior.

4.2 Games with different information

Five reasonable and different games are described in this section, where the basic control structure of Fig. 4.1 is used as basis structure for comparison. The second, third and fourth controller structures are modified in such a way that the control system contains different control laws, leading to different information sets of the players.

A further differential game is considered here, which is not mentioned previously in this work, where the order of decision making or strategy selection is considered. The strategies of the players are, among others, dependent on the controller parameters Q_{ij} and P_{ij}. In this game, the controller parameters Q_{ij} and P_{ij} are not tuned, according to a multiple parameter optimization at the same time. In contrast, the controller parameters Q_{ij} and P_{ij} of the leader (player i) are optimized first. Dependent on the parameter set Q_{ij} and P_{ij}, the parameters $Q_{\neg ij}$ and $P_{\neg ij}$ of the other player $\neg i$ are optimized. In the previous approach of the control system design, the author considered the parameter optimization at the same time. In contrast, the basic control structure studied in this part allows the additional consideration of a leader-follower game, which is also known in the literature as Stackelberg game.

The forthcoming descriptions of the varying games are restricted through those components which distinguish the games from each other. These components are the information sets of both players containing the error signals $E_i(s)$, with steps $(1/s)$ as references R_i, needed for the calculation of the cost functions J_{cf_i}.

For shortage of space, the polynomials $A_{ij}(s)$, $B_{ij}(s)$, $P_{ij}(s)$, $Q_{ij}(s)$, $E_i(s)$ and $R_i(s)$ are abbreviated in the following as A_{ij}, B_{ij}, P_{ij}, Q_{ij}, E_i, and R_i.

4.2.1 Game I - the basic game

The first game (GI) is played based on the given control structure in Fig. 4.1. The information set of player 1 consists of the error signal E_1, dependent on the controller parameter set Q_{11} and P_{11}, and the control law of the upper control loop.

In contrast, the information set of player 2 includes his own control law, and his controller parameter sets Q_{22} and P_{22} as well as the control law, and the controller parameters Q_{11} and P_{11} of player 1. According to the expressions of Section 4.1.2, the information structure of game I is incomplete and asymmetric.
To obtain a fast reference tracking with low deviation, the required error functions E_1 and E_2 of game I are formulated as:

$$E_1 = \frac{A_{11}P_{11}R_1}{A_{11}P_{11} + B_{11}Q_{11}} \tag{4.6}$$

for the first player, and

$$E_2 = \frac{A_{21}A_{22}P_{22}(A_{11}P_{11} + B_{11}Q_{11})R_2 - B_{21}Q_{11}A_{11}A_{22}P_{22}R_1}{A_{21}(A_{11}P_{11} + B_{11}Q_{11})(A_{22}P_{22} + B_{22}Q_{22})} \tag{4.7}$$

for the second player.
As a result, the cost function J_{cf_1} for player I contains only elements of the upper control loop while the cost function J_{cf_2} of the second player consists of elements of the lower control loop as well as elements of the upper control loop of player I.

4.2.2 Game II - a game with forward information flow

In the game with a forward information flow, no modifications on the information sets of both players are made. Only an additional controller is inserted to be able to add the error signal E_1 of the first player to the control signal U_2 of the second one as displayed in Fig. 4.2. Hence, player 2 gets information about player $1's$ input earlier. This makes it possible for him to react earlier on potential disturbances.
The additional controller C_{21}, together with controller C_{22}, represents the second player, choosing the strategy u_2. The error functions E_1, and E_2 of game II (GII) are

$$E_1 = \frac{A_{11}P_{11}R_1}{A_{11}P_{11} + B_{11}Q_{11}} \tag{4.8}$$

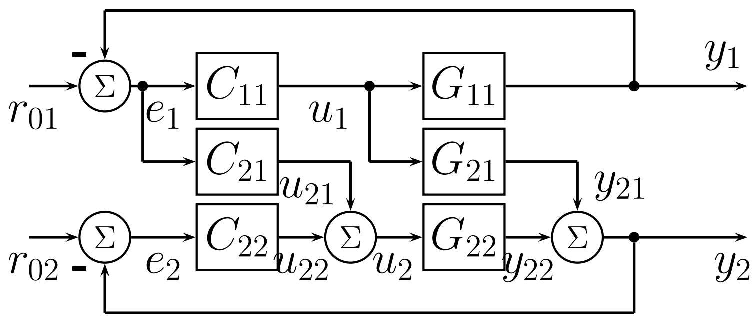

Figure 4.2. Control structure of game II and game III.

for the first player, and

$$E_2 = \frac{(A_{11}P_{11} + B_{11}Q_{11})A_{21}A_{22}P_{21}P_{22}R_2}{A_{21}P_{21}(A_{22}P_{22} + B_{22}Q_{22})(A_{11}P_{11} + B_{11}Q_{11})} - \frac{(B_{21}Q_{11}A_{22}P_{21} + B_{22}Q_{21}A_{21}P_{11})A_{11}P_{22}R_1}{A_{21}P_{21}(A_{22}P_{22} + B_{22}Q_{22})(A_{11}P_{11} + B_{11}Q_{11})} \tag{4.9}$$

for the second player.
Like in the game before, the cost function J_{cf_1} for player 1 contains only elements of the upper control loop while the cost function J_{cf_2} of the second player consists of elements included in the lower control loop as well as elements of the upper control loop of player 1. There is no modification on the information sets for player 1 or player 2.

4.2.3 Game III - a game with a decoupler

Game III (GIII) plays with the same control structure as game II, displayed in Fig. 4.2 with the distinction that controller C_{21} now acts as a decoupler with

$$C_{21} = -\frac{G_{21}}{G_{11}} = -\frac{B_{21}A_{11}}{A_{21}B_{11}} \tag{4.10}$$

to reformulate

$$G = \begin{bmatrix} G_{11} & 0 \\ G_{21} & G_{22} \end{bmatrix} \tag{4.11}$$

as

$$G^* = \begin{bmatrix} 1 & 0 \\ C_{21} & 1 \end{bmatrix} \cdot \begin{bmatrix} G_{11} & 0 \\ G_{21} & G_{22} \end{bmatrix} = \begin{bmatrix} G_1 & 0 \\ 0 & G_2 \end{bmatrix}. \tag{4.12}$$

The intention of a decoupler is to mathematically eliminate the effect of interactions in transforming the process matrix into a diagonal matrix using a transfer matrix. Hence, no parameters has to be tuned for this controller C_{21} and the error signals E_1 and E_2 are formulated like (4.8) and (4.9), except that $Q_{21} = -B_{21}A_{11}$ and $P_{21} = A_{21}B_{11}$.
However, this approach is limited through basics of control theory. The decoupling method could translate zeros to poles and unstable decoupling elements may result.

4.2.4 Game IV - a game with complete information

The most interesting game in the perspective of game theory is the game (GIV) where a modification of one information set is made. Thus, an additional controller C_{12} is added to the structure, see Fig. 4.3, adding the error signal E_2 of the second player to the control signal U_1 of player 1.

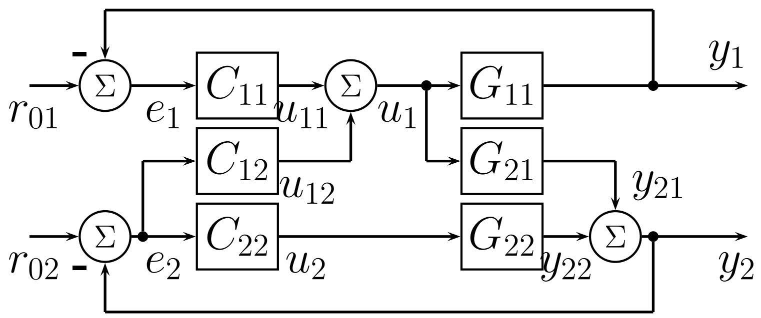

Figure 4.3. Control structure of game IV.

Player 1 extends its information set with the control law and the controller parameters Q_{22} and P_{22} of player 2, while the information set for the second player remains unchanged. The information structure is changed resulting in a game with complete information. All three controllers C_{11}, C_{12}, and C_{22} are tuned while minimizing the cost functions J_{cf_i} of both players. The corresponding error signals

E_1 and E_2 needed to calculate the costs for a satisfying reference tracking are

$$\begin{aligned} E_1 = & \frac{(A_{21}P_{12}A_{22}P_{22} + B_{21}Q_{12}A_{22}P_{22} + B_{22}Q_{22}A_{21}P_{12})A_{11}P_{11}R_1}{T} \\ & - \frac{B_{11}Q_{12}P_{11}A_{21}A_{22}P_{22}R_2}{T} \end{aligned} \tag{4.13}$$

and

$$\begin{aligned} E_2 = & \frac{(A_{11}P_{11} + B_{11}Q_{11})A_{21}P_{12}A_{22}P_{22}R_2}{T} \\ & - \frac{B_{21}Q_{11}A_{11}P_{12}A_{22}P_{22}R_1}{T} \end{aligned} \tag{4.14}$$

with $T = (A_{11}P_{11} + B_{11}Q_{11})(A_{21}P_{12}A_{22}P_{22} + B_{21}Q_{12}A_{22}P_{22} + B_{22}Q_{22}A_{21}P_{12}) - B_{11}Q_{12}B_{21}Q_{11}A_{22}P_{22}$.
Considering the elements of the error functions E_1 and E_2 in (4.13) and (4.14), one can conclude that there exist an analogy to the information sets of the players. Equations (4.13), and (4.14) both contain elements, or information, of the other players.

4.2.5 Game V - a Stackelberg game

According to (Basar u. Olsder, 1999), games, in which one player, called the leader, declares his strategy first and enforces it on the other player, called the follower, is called a Stackelberg game.
As in the given basic triangular control structure of Fig. 4.1 there is only one connection from player 1 to player 2, while player 2 has no effect on the control loop of player 1 at all. This leads to the idea that player 1 can be treated as a *leader* and player 2 as a *follower* in a Stackelberg game. The main advantage of this approach is that no trade off has to be met.
Player 1 minimizes his cost J_{cf_1} while choosing an optimal parameter set Q_{11} and P_{11}. Under consideration of the resulting strategy u_1, player 2 chooses his parameter set Q_{22} and P_{22} depending on the minimization of his cost and the use of the leader's strategy u_1, including the parameter set Q_{11} and P_{11}. Concerning game V (GV), the corresponding error functions E_1 and E_2 are evaluated as in (4.6) and (4.7) but with the attention of the order in decision making.

Chapter 5

Case Study 1: A Two-input/Two-output Differential Game

The application of a reverse osmosis (RO) desalination plant is used as an example of a MIMO system, with a 2×2 control structure. This application is chosen due to the only one-way interaction existing in the system. Using such a triangular control system structure, one control loop is unaffected, while the second is disturbed through the first mentioned. Applying the proposed method to a one-way interacting control system is advantageous for the study of the control behavior improvement, compared to conventional tuning methods. The advantage is, that the second control loop is only disturbed by the first one and not through itself as it is the case in a cross coupling system. The application of the developed approach on a 2×2 cross coupled system as well as a cascade control system structure is shown in App. A.

For the continuous implementation of the reverse osmosis system, the requirements are set on a fast reference tracking with low deviation and low control effort in both control loops. Additionally, in a second release, the robustness of the system is studied.

5.1 Multi-loop control system design for MIMO systems: The Reverse Osmosis Desalination Plant

5.1.1 Example Description

The ultimate ambition of a RO desalination process is producing a constant quantity of water with an acceptable purity. In this context, several papers were published, for example (Assef u. a., 1995), (Riverol u. Pilipovik, 2005), (Gambier u. a., 2006) and (Robertson u. a., 1996), where RO system identification is considered as a two-input/two-output (TITO) system. The two input variables are the transmembrane pressure (P) and the feed pH (pH), whereas the controlled output variables are the permeate flux (F) and the permeate conductivity (C). The system interaction can be written as

$$\begin{bmatrix} \mathrm{F} \\ \mathrm{C} \end{bmatrix} = \begin{bmatrix} G_{p11} & G_{p12} \\ G_{p21} & G_{p22} \end{bmatrix} \begin{bmatrix} \mathrm{P} \\ \mathrm{pH} \end{bmatrix}, \tag{5.1}$$

belonging to the control structure of Fig. 5.1.

The control structure of Fig. 5.1 consists of two controllers C_1 and C_2 and four

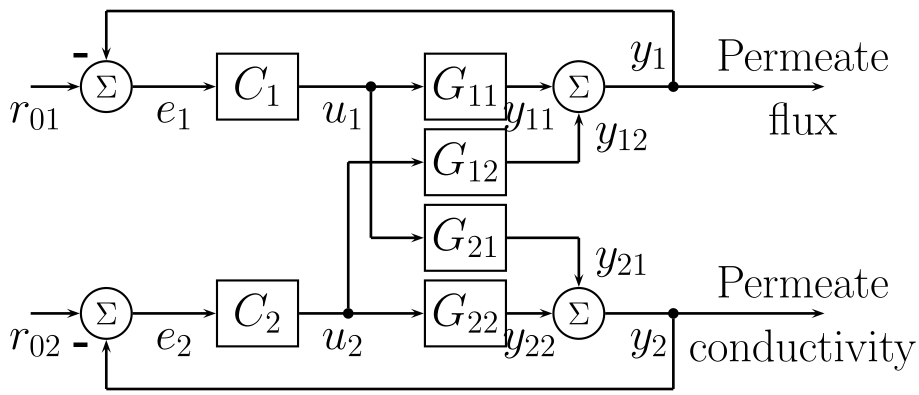

Figure 5.1. Control structure of the RO desalination process.

process transfer functions G_{11}, G_{12}, G_{21} and G_{22}. The sum of the process transfer functions G_{11} and G_{12} provides the permeate flux, while the sum of the process

transfer functions G_{21} and G_{22} forms the permeate conductivity.
The process transfer functions, used in this work are chosen from (Robertson u. a., 1996), relating the inputs to the outputs as follows:

$$\frac{F}{P} = G_{p11} = \frac{B_{11}}{A_{11}} = \frac{0.002(0.056s + 1)}{(0.003s^2 + 0.1s + 1)} \tag{5.2}$$

$$\frac{F}{pH} = G_{p12} = \frac{B_{12}}{A_{12}} = 0 \tag{5.3}$$

$$\frac{C}{P} = G_{p21} = \frac{B_{21}}{A_{21}} = \frac{-0.51(0.35s + 1)}{(0.213s^2 + 0.7s + 1)} \tag{5.4}$$

$$\frac{C}{pH} = G_{p22} = \frac{B_{22}}{A_{22}} = \frac{-57(0.32s + 1)}{(0.6s^2 + 1.8s + 1)} \tag{5.5}$$

In words, a change in the transmembrane pressure (P) effects the permeate flux as well as causing a negative effect on the permeate conductivity (C). Changing the pH has no effect on the permeate flux (F), due to (5.3), but it causes a negative effect in the permeate conductivity (C).
The control structure reflects the triangular (asymmetric) dependency in such a way that the upper control loop acts as a disturbance on the lower control loop. Thus, the control loops of the multi-loop system interact only one-way, compare section 4, Fig. 4.1.
The operating point of the desalination plant is given in Table 5.1.

The control system design with optimal performance concerning the reference

Table 5.1. Operating point of the RO desalination process.

Variable	Linear range
Flux [gpm]	0.85-1.25
Pressure [psi]	800-1000
Conductivity [μS/cm]	400-450
pH	6-7

tracking and the control effort is now implemented using the proposed game-theoretic approach.

5.2 Multi-loop control system design for the continuous reverse osmosis desalination system

For the continuous implementation of a MIMO reverse osmosis process, the game-theoretic approach is applied. It is composed of a game description, a cost function set up, description of the course of the game and the final solution selection.

5.2.0.1 Game description

The differential game description of the reverse osmosis plant is modelled by the coprime rational expressions

$$\frac{Y_{11}(s)}{U_1(s)} = G_{11}(s) = \frac{B_{11}(s)}{A_{11}(s)} \tag{5.6}$$

$$\frac{Y_{12}(s)}{U_2(s)} = G_{12}(s) = \frac{B_{12}(s)}{A_{12}(s)}, \tag{5.7}$$

$$\frac{Y_{21}(s)}{U_1(s)} = G_{21}(s) = \frac{B_{21}(s)}{A_{21}(s)}, \tag{5.8}$$

and

$$\frac{Y_{22}(s)}{U_2(s)} = G_{22}(s) = \frac{B_{22}(s)}{A_{22}(s)}. \tag{5.9}$$

The control laws of both control loops are given by

$$U_1(s) = C_1(s)E_1(s) = \frac{Q_1(s)}{P_1(s)}E_1(s) \tag{5.10}$$

and

$$U_2(s) = C_2(s)E_2(s) = \frac{Q_2(s)}{P_2(s)}E_2(s). \tag{5.11}$$

The polynomial descriptions of the PI controllers C_1 and C_2 with proportional parameters K_{P_1}, K_{P_2} and integral parameters K_{T_1}, K_{T_2} are

$$C_1 = \frac{Q_1}{P_1} = \frac{K_{P_1}s + K_{P_1}/K_{T_1}}{s} \tag{5.12}$$

and

$$C_2 = \frac{Q_2}{P_2} = \frac{K_{P_2}s + K_{P_2}/K_{T_2}}{s}. \tag{5.13}$$

The control system design of the TITO system in Fig. 5.1 is considered as a differential game between two players on the time period $[0, \infty]$.
The strategies of the players are defined as

$$u_1(t) = \int_0^\infty c_1(t)e_1(t-\tau)d\tau \tag{5.14}$$

and

$$u_2(t) = \int_0^\infty c_2(t)e_2(t-\tau)d\tau \tag{5.15}$$

with

$$\mathcal{L}\{c_1(t)\} = C_1(s) = \frac{Q_1}{P_1} = \frac{K_{P_1}s + K_{P_1}/K_{TI_1}}{s} \tag{5.16}$$

and

$$\mathcal{L}\{c_2(t)\} = C_2(s) = \frac{Q_2}{P_2} = \frac{K_{P_2}s + K_{P_2}/K_{TI_2}}{s}. \tag{5.17}$$

Q_1 and Q_2 are polynomials and contain the proportional and integral controller parameters of C_1 and C_2. The strategies u_i of the players belong to the strategy sets $\mathcal{U}_i = \{u_i | u_i \text{ is given by (5.14) and (5.15)}\}$.
The differential game is now described as the evolution of the errors e_i with

$$e_1^{(6)}(t) = f(e_1^{(5)}(t), e_1^{(4)}(t), e_1^{(3)}(t), \ddot{e}_1(t), \dot{e}_1(t), u_1, u_2), \tag{5.18}$$

and

$$e_2^{(6)}(t) = f(e_2^{(5)}(t), e_2^{(4)}(t), e_2^{(3)}(t), \ddot{e}_2(t), \dot{e}_2(t), u_1, u_2) \tag{5.19}$$

and the initial conditions $e_1(0) = e_{10}$ and $e_2(0) = e_{20}$. The errors e_1 and e_2 belong to the sets $E_1 = \{e_1 | e_1 \text{ as solution of (5.18)}\}$ and $E_2 = \{e_2 | e_2 \text{ as solution of (5.19)}\}$, respectively.
Function f_1 is defined on $f_1 : R_1 \times \mathcal{U}_1 \to \mathbb{R}^+$ and $f_2 : R_1 \times R_2 \times \mathcal{U}_1 \times \mathcal{U}_2 \to \mathbb{R}^+$.
In the first release of the control system design for the reverse osmosis system, the specifications are set on a fast reference tracking with low deviation and low control effort, described, in detail, in the upcoming subsection.

5.2.1 Multi-loop control system design subject to a fast reference tracking with low deviation and low control effort

5.2.1.1 Cost function and constraint set up

The requirements on a satisfying reference tracking is implemented using the appropriate cost functions, proposed in Section 3.2.1 for the Integral Square Error (ISE) in (3.23).
According to Section 3.2.2, the requirement on a low control effort is realized - for comparison - in four different ways:

(A) Add on to existing cost functions (3.23) using a Lagrangian multiplier λ. The addition of the control constraints to the cost function implementations, concerning the reference reaction using a Lagrangian multiplier provides cost functions as formulated in (3.27).

(B) Control effort as cost function implementation. The requirement on low control effort is implemented in considering the square of the control signals derivative separately. This formulates the requirement as single cost functions.

(C) Explicit control constraints implementation. Since every control signal cannot be followed by the physical system, the control signals $u_i(t)$ of the plant are limited around the operating point in a predefined range of $\pm L_i$:

$$|u_1(t)| \leq L_1 \text{ and } |u_2(t)| \leq L_2. \tag{5.20}$$

These constraints are kept during the course of the game with the present challenge to optimize (3.23) subject to (5.20).

(D) Explicit control constraints as cost function implementation. Additionally, the idea is to keep the deviation of the control signals to the linearization point as small as possible. To obtain the smallest possible control signal deviation, the distance between the control signals $u_1(t)$, $u_2(t)$ and the limits

$\pm L_1$, $\pm L_2$ should be as large as possible. Also, these constraints can be reformulated as

$$max(L_1 - |u_1(t)|) \tag{5.21}$$

and

$$max(L_2 - |u_2(t)|). \tag{5.22}$$

Since every objective function is minimized in this work, (5.21) and (5.22) are rewritten as objective functions J_{u1}, J_{u2} for the control signals as

$$J_{u1} = -(L_1 - |u_1(t)|), \tag{5.23}$$

and

$$J_{u2} = -(L_2 - |u_2(t)|). \tag{5.24}$$

In this case each player has to satisfy two objective functions, and the challenge is now, to optimize (3.23) as well as (5.23) and (5.24) simultaneously.

Derived from the control structure of Fig. 5.1, the plant model of (5.2)-(5.5) and the control law given by (5.14) and (5.15), the transfer function for the control signal $E_1(s)$ of the first loop is given as

$$E_1(s) = \frac{A_{11}P_1}{A_{11}P_1 + B_{11}Q_1}R_1. \tag{5.25}$$

The corresponding control signal $U_1(s)$ is formulated as

$$U_1(s) = \frac{Q_1A_{11}}{A_{11}P_1 + B_{11}Q_1}R_1(s). \tag{5.26}$$

For the second control loop, the error signal $E_2(s)$ and the control signal $U_2(s)$ is

$$\begin{aligned} E_2(s) =& \frac{A_{22}P_{22}}{A_{22}P_2 + B_{22}Q_2}R_2 \\ &- \frac{B_{21}Q_1A_{11}A_{22}P_2}{A_{21}(A_{11}P_1 + B_{11}Q_1)(A_{22}P_2 + B_{22}Q_2)}R_1 \end{aligned} \tag{5.27}$$

and

$$
\begin{aligned}
U_2(s) = & \frac{Q_1 A_{22}}{A_{22}P_2 + B_{22}Q_2} R_2 \\
& - \frac{Q_2 B_{21} Q_1 A_{11} A_{22}}{(A_{11}P_1 + B_{11}Q_1)(A_{22}P_2 + B_{22}Q_2)} R_1,
\end{aligned}
\tag{5.28}
$$

respectively.

Concerning a satisfying reference tracking, equations (5.25) and (5.27) are used to calculate the objective functions for the control error signals J_{e1} and J_{e2} of (3.23). Considering the requirement on low control effort, the four different implementations are studied. For the cost functions derivative the relation $\dot{u}(t) \circ\!\!-\!\!\bullet\, sU(s)$ is applied.

5.2.1.2 A) Control effort added to existing cost functions

In the first implementation of a low control error demand, equation (3.27) is applied as cost function for each player.

Obtaining the Pareto-optimal set and the final solution

An abstract of the chosen GA parameter settings is listed in Table 5.2. The genetic algorithm operates with 100 generations and 4 chromosomes, representing the controller parameters, two for each controller. Two subpopulations with 100 individuals each are chosen and the number of cost functions is 2.

Table 5.2. Algorithms and parameters for the MOO.

Evolutionary algorithm	Values
Number of generations	100
Number of chromosomes	4
Subpopulations	2
Individuals (at start per subpopulation)	100 100
Number of objective functions	2
Selection pressure gen. gap	Stochastic universal sampling 2.1 0.9
Reinsertion rate	Local reinsertion 1
Recombination rate	Discrete recombination 1
Mutation rate range	Real valued mutation 1 0.1

Using the proposed approach, controllers are designed, where the parameter vector $\boldsymbol{\chi}_A$ for the players is of the form

$$\boldsymbol{\chi}_A = [K_{P1},\ K_{I1},\ K_{P2},\ K_{I2}]\,, \tag{5.29}$$

with $K_{Ii} = K_{Pi}/K_{Ti}$, providing the four chromosomes for the GA.

The controller parameters for a reference case are tuned according to a modified Ziegler-Nichols method given in (Robertson u. a., 1996) and are listed in Table 5.3.

Table 5.3. PI controller parameters using the modified Ziegler-Nichols tuning method.

Controller i	K_{Pi}	K_{Ti}
C_1	536	0.23
C_2	-0.05	1.81

The range for the tuning parameters is kept around the parameters of the reference case, see Table 5.4, and is set to

$$\begin{aligned} 1 \leq K_{P1} &\leq 1000 \\ 1 \leq K_{I1} &\leq 10000 \\ -10 \leq K_{P2} &\leq -0.001 \\ -10 \leq K_{I2} &\leq -0.001. \end{aligned} \tag{5.30}$$

The negative parameters of the second controller are justified through the negative transfer function G_{22}, which is applied on the lower control loop. As both parameters K_{P2} and K_{I2} are negative, the resulting reset time K_{T2} is positive.
To study the influence of the Lagrangian multiplier λ_i, three different settings for λ_i are implemented. First, $\lambda_i = 1$ with complete control effort consideration, second, $\lambda_i = 0.25$ with only few control effort consideration and third, $\lambda_i = 0$ with no control effort consideration during the control system design. The ISE cost function implementations is applied and the controller parameters are obtained using the Nash bargaining solution concept. The corresponding controller parameters are listed in Tab.5.4.

Concerning the number of non dominated solutions, their quantity increases as

Table 5.4. A) Controller parameters K_{P1}, K_{I1}, K_{P2} and K_{I2} for the continuous reverse osmosis system according to low control effort, which is added to cost functions concerning the reference tracking.

case	K_{P1}	K_{I1}	K_{P2}	K_{I2}	# *nondom*
RefCase	536	2330.435	−0.05	−0.028	–
$\lambda_i = 1$	1.3902	137.7051	−2.8243	−8.8546	13
$\lambda_i = 0.25$	1	1416.4	−1.783	−10	12
$\lambda_i = 0$	189.1545	9875	−10	−9.9941	88

λ_i gets closer to or equal to zero.
The obtained parameters for the different games vary within the predefined range of (5.30).

Simulation results

The operating point of the plant is given by a permeate flux of 0.85 gpm ($0.2\,{}^{m^3}/_{\mathrm{h}}$) and a conductivity of $400\,{}^{\mu\mathrm{S}}/_{\mathrm{cm}}$. Fig. 5.2 shows the response of a 0.4 gpm step change to 1.25 gpm ($0.3\,{}^{m^2}/_{\mathrm{h}}$) in the set point of the permeate flux.

Considering the step responses of the permeate flux the step responses for $\lambda_i = 0$

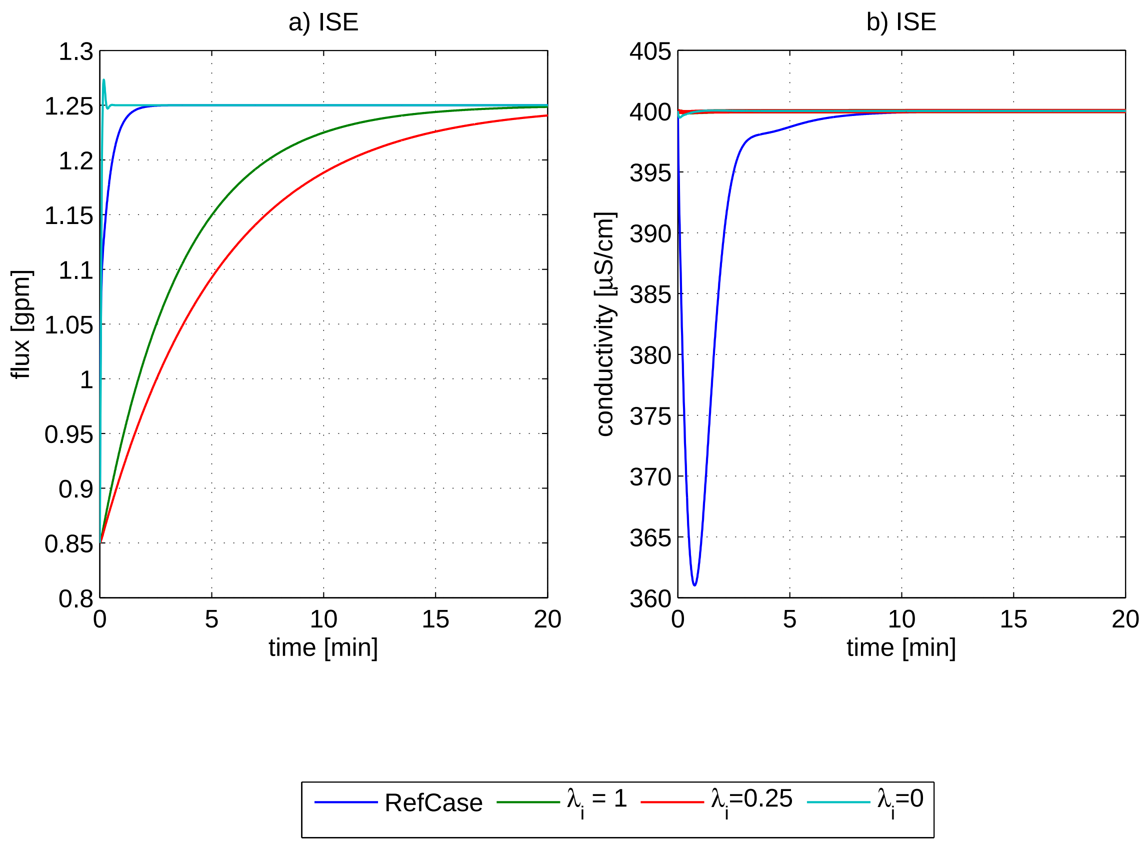

Figure 5.2. A) Output responses to a change in the set point of the permeate flux according to low control effort, which is added to cost functions concerning the reference tracking.

shows faster set point convergence compared to the reference case but with overshoot. Concerning the corresponding step responses for $\lambda_i = 1$ and $\lambda_i = 0.25$, the set point convergences are slower and respectively equal in their behavior, compared to the reference case. The faster reference tracking for $\lambda_i = 1$ compared to $\lambda_i = 0.25$ is explainable with their size of the solution sets. As given in Tab.5.4, the number of Pareto-optimal points for $\lambda_i = 1$ is 13 and for $\lambda_i = 0.25$ it is 12. So both solution sets does not have such a high resolution than $\lambda_i = 0$ with 88 non dominated solutions. Applying the final solution concept on such small solution sets could result in a better performance for games that should show a slower set point convergence and larger control signals.

The step responses of the conductivity according to a change in the set point of the flux are shown in the subplot *b*) in Fig. 5.2. The caused disturbance in the conductivity of the reference case is immense. It is about 10 % of the set point and around 9 minutes are required for it's compensation. These disturbances are incommensurated to those of the reference case. The step responses of the game-theoretic controller tuning method show only small disturbances, compared to the reference case and independent of the value for λ_i.

Fig. 5.3 shows, that a change in the conductivity set point, from 400 $^{\mu S}/_{cm}$ to 410 $^{\mu S}/_{cm}$, has no influence on the flux due to the triangular structure. Concerning the conductivity, all cases, except the reference case of (Robertson u. a., 1996) shows a fast set point convergence response with fast rise time and only small overshoot.

For all values of λ_i, the corresponding error signals and control signals are displayed in Fig. 5.4 and Fig. 5.5, respectively.

The subplots *a*) and *b*) of Fig. 5.4 show the error signals e_1 and e_2 as a result of a step in the permeate flux. Comparable with the plots in Fig. 5.3, the reference case shows the largest error in subplot *b*). While the error size of e_1 is about 0.4 gpm (which is equal to the step size and the largest error for $\lambda_i = 0$).

The subplots *c*) and *d*) of Fig. 5.4 show the error signals due to a step in the conductivity with no effect in e_1 as a result of the triangular structure. Concerning e_2, the reference case needs about 9 minutes for the error convergence, and the others converge within the first minute.

Concerning the control signals in Fig. 5.5, subplots *a*) and *b*) show the corresponding control signals to a step in the permeate flux, and subplots *c*) and *d*) show the corresponding control signals to a step in the conductivity. Considering u_1 in subplot *a*), the amplitudes are of comparable size, as the control signal moves

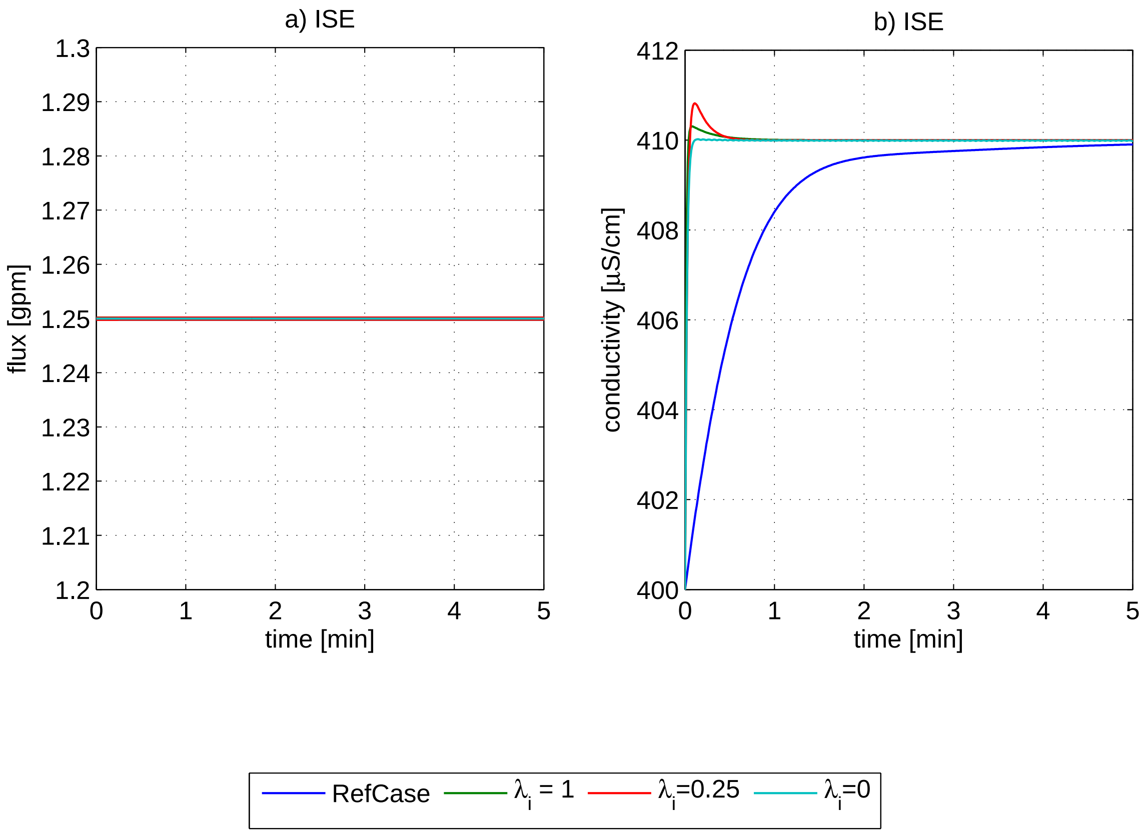

Figure 5.3. A) Output responses to a change in the set point of the conductivity according to low control effort. This is added to cost functions concerning the reference tracking.

from 800 psi to 1000 psi and all displayed control signals moves within this range. Except for $\lambda_i = 0$ a small overshoot is given in u_1. In contrast, in subplots b) and d), the influence of the value of λ_i is visible: for $\lambda_i = 0$, the corresponding amplitudes are larger (up to five times) than for $\lambda_i \neq 0$, while the reference case shows the smallest amplitudes. In fact, for a step of 10 $^{\mu S}/_{cm}$ in the conductivity, the control signal u_2 moves from 6 pH to 5.83 pH. A step of 0.4 gpm in the flux causes a change in the control signal u_2 from 6 pH to 4.21 pH. This is to compensate for the caused error in conductivity.

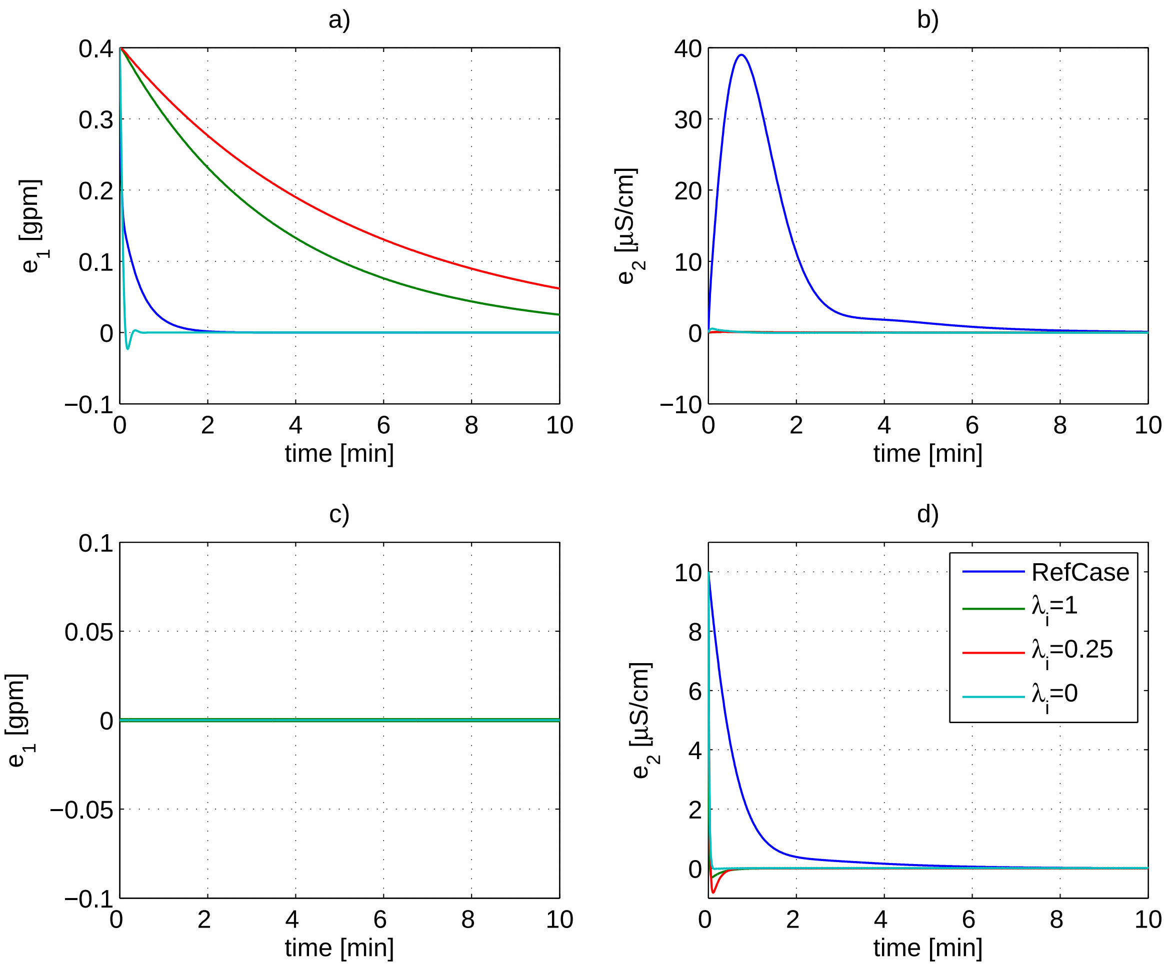

Figure 5.4. A) Error signals for the ISE implementation with varying λ_i according to a set point change in the permeate flux (subplots a) and b)) and a set point change in the permeate conductivity (subplots c) and d)).

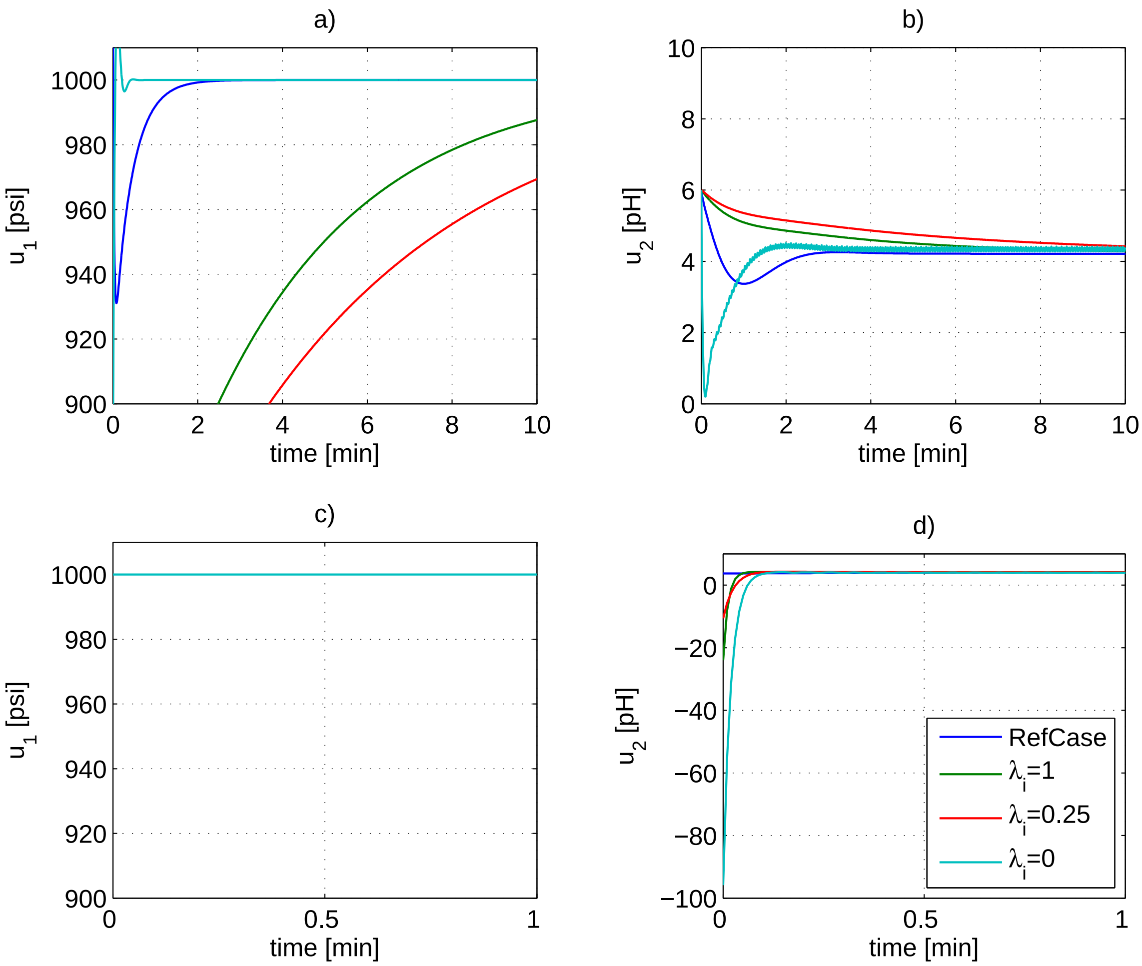

Figure 5.5. A) Control signals for the ISE implementation with varying λ_i according to a set point change in the permeate flux (subplots a) and b)) and a set point change in the permeate conductivity (subplots c) and d)).

5.2.1.3 B) Control effort as cost function implementation

For the implementation of two additional cost functions, the genetic algorithm operates with 100 generations and 4 chromosomes, two for each controller. Two subpopulations with 100 individuals each are chosen, and the number of cost functions is changed to 4.

Obtaining the Pareto-optimal set and the final solution
Using the proposed approach, controllers are designed. The parameter vector $\boldsymbol{\chi}_B$ for the players is of the form

$$\boldsymbol{\chi}_B = [K_{P1},\ K_{I1},\ K_{P2},\ K_{I2}]\,, \tag{5.31}$$

with $K_{Ii} = K_{Pi}/K_{Ti}$, providing the four chromosomes for the GA. The corresponding ranges are set according to (5.30). Again, three different cost function implementations are studied.
The Nash bargaining solutions, providing the respective controller parameter sets, are listed in Table 5.5.

Table 5.5. B) Controller parameters K_{P1}, K_{I1}, K_{P2} and K_{I2} for the continuous reverse osmosis system according to the reference tracking and low control effort, formalized as cost functions.

case	K_{P1}	K_{I1}	K_{P2}	K_{I2}	$\#$ *nondom*
$RefCase$	536	2330.435	-0.05	-0.028	$-$
ISE	110.7332	9865.2	-2.4053	-9.8672	180

Simulation results
The step sizes are chosen comparable to section 5.2.1.2. The appropriate simulation results to a step in the flux are shown in Fig. 5.6. The step responses for the ISE converge faster to the set point in comparison to the reference case. Again, the introduced error in the control loop for the conductivity is of larger size for the reference case, compared to the game-theoretic solution. Concerning the subplots displaying the control signals, the ISE implementation shows overshoots in u_1 and the reference case shows overshoot in u_2.

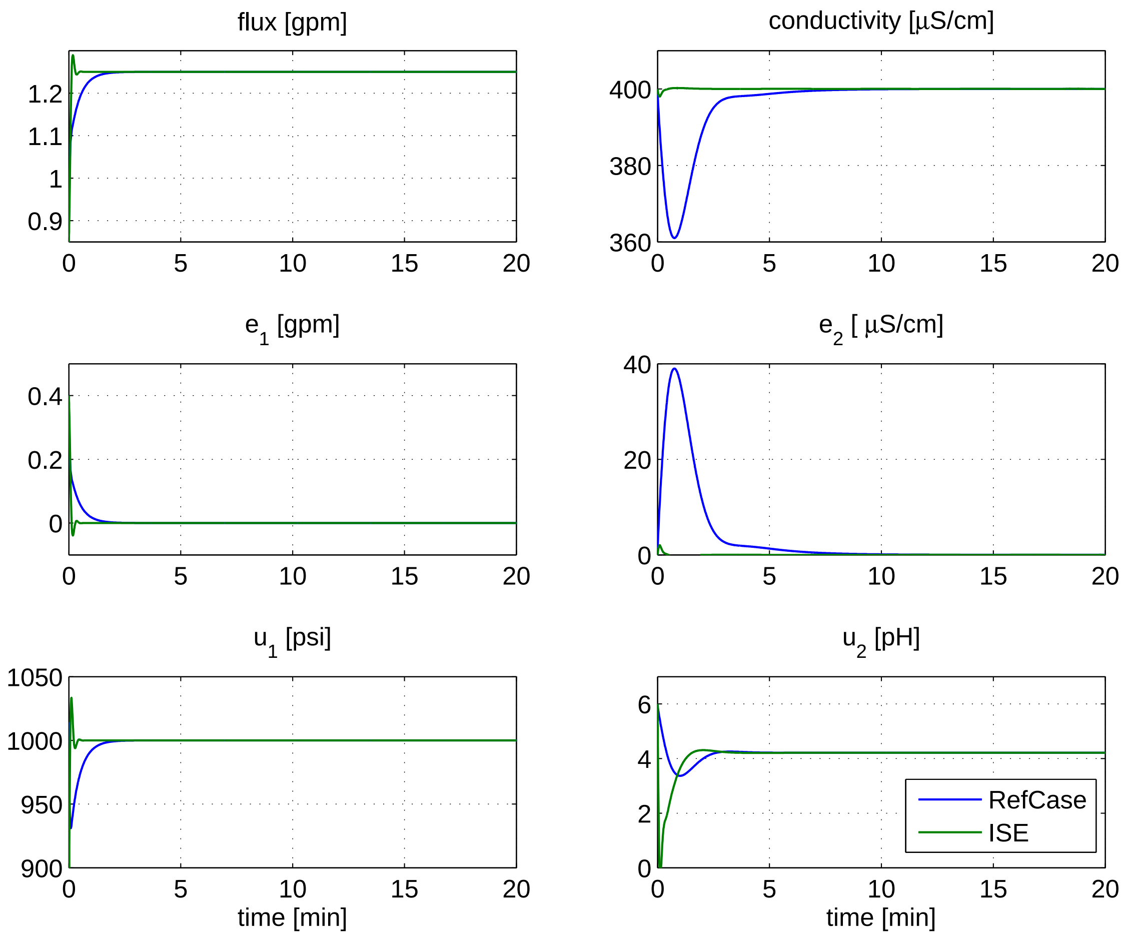

Figure 5.6. B) Responses to a change of 0.4 gpm in the permeate flux according to the reference tracking and low control effort, formalized as cost functions.

In Fig. 5.7, the simulation results are shown as result of a step in the conductivity. Once again, with no effects in the components of the upper control loop. The error convergence of the second control loop (compare the subplots for the conductivity and e_2) behave faster than the reference case. Meanwhile, the corresponding control signal u_2 shows more overshoot than the reference case.

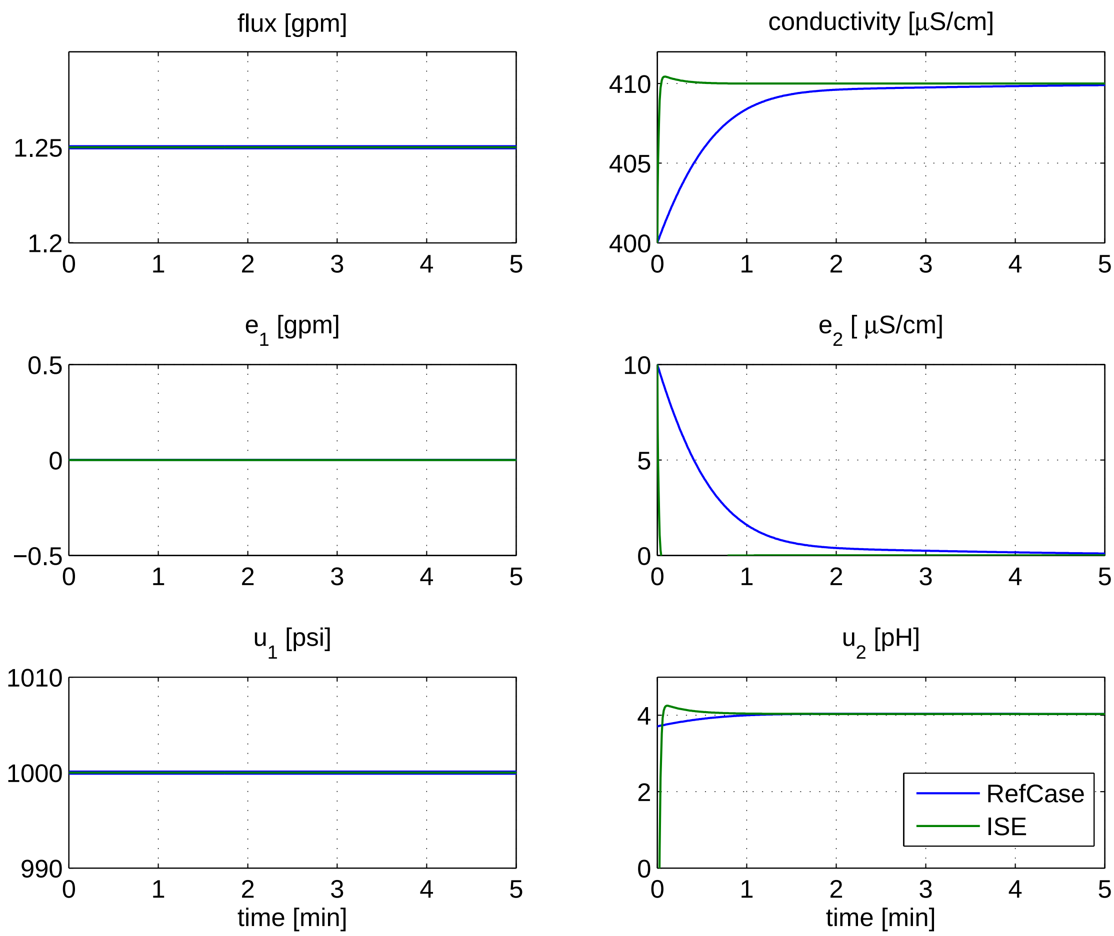

Figure 5.7. B) Response to a change of 10 μS in the conductivity according to the reference tracking and low control effort, formalized as cost functions.

5.2.1.4 C) Explicit control constraints implementation

In the third implementation, explicit constraints L_1 and L_2 for the RO desalination process are required. Different constraints L_1 and L_2 on the controls $u_1(t)$ and $u_2(t)$ of the RO process are assigned in the subsequent section. The settings of the genetic algorithm remains, except the number of cost functions is changed to 2.

Obtaining the Pareto-optimal set and the final solution

Using the proposed approach, controllers are designed. The parameter vector $\boldsymbol{\chi}_C$ for the players is of the form

$$\boldsymbol{\chi}_C = [K_{P1},\ K_{I1},\ K_{P2},\ K_{I2}]\,, \tag{5.32}$$

with $K_{Ii} = K_{Pi}/K_{Ti}$, providing the four chromosomes for the GA. Once more, the corresponding ranges are set according to (5.30), and the ISE cost function implementations are applied. The implemented constraints on the control signals accept an overshoot on the control signals of 10 %, 5 % and 1 %, according to a step change in the set points.
The corresponding Nash bargaining solutions, providing the respective controller parameter sets, are listed in Table 5.6.

Table 5.6. C) Controller parameters K_{P1}, K_{I1}, K_{P2} and K_{I2} for the continuous reverse osmosis system according to explicit control constraints.

case	overshoot	K_{P1}	K_{I1}	K_{P2}	K_{I2}	*# nondom*
RefCase		536	2330.435	−0.05	−0.028	–
ISE						
	10 %	114.1680	10000	−10	−10	80
	5 %	226.3851	3626.7	−10	−9.8779	39
	1 %	288.8275	1166.4	−9.9922	−9.7285	28

Studying Table 5.6, it is noticeable that, the constraints on the control signals has effects on the number of non dominated points. The number of non dominated points decrease the smaller, and the permitted amplitudes of the control efforts are chosen.

Simulation results

Again, the step sizes are chosen as in section 5.2.1.2. The step responses of the permeate flux and the permeate conductivity to a step in the flux are displayed in Fig. 5.8. The response of the flux, accepting only 1 % overshoot in the control signals show slowlier set point convergence compared to the reference case. The others are faster. Accepting a 10 % overshoot in the control signals even generates an overshoot in the flux response before set point convergence.

The step responses of the flux and the conductivity to a change in the set point

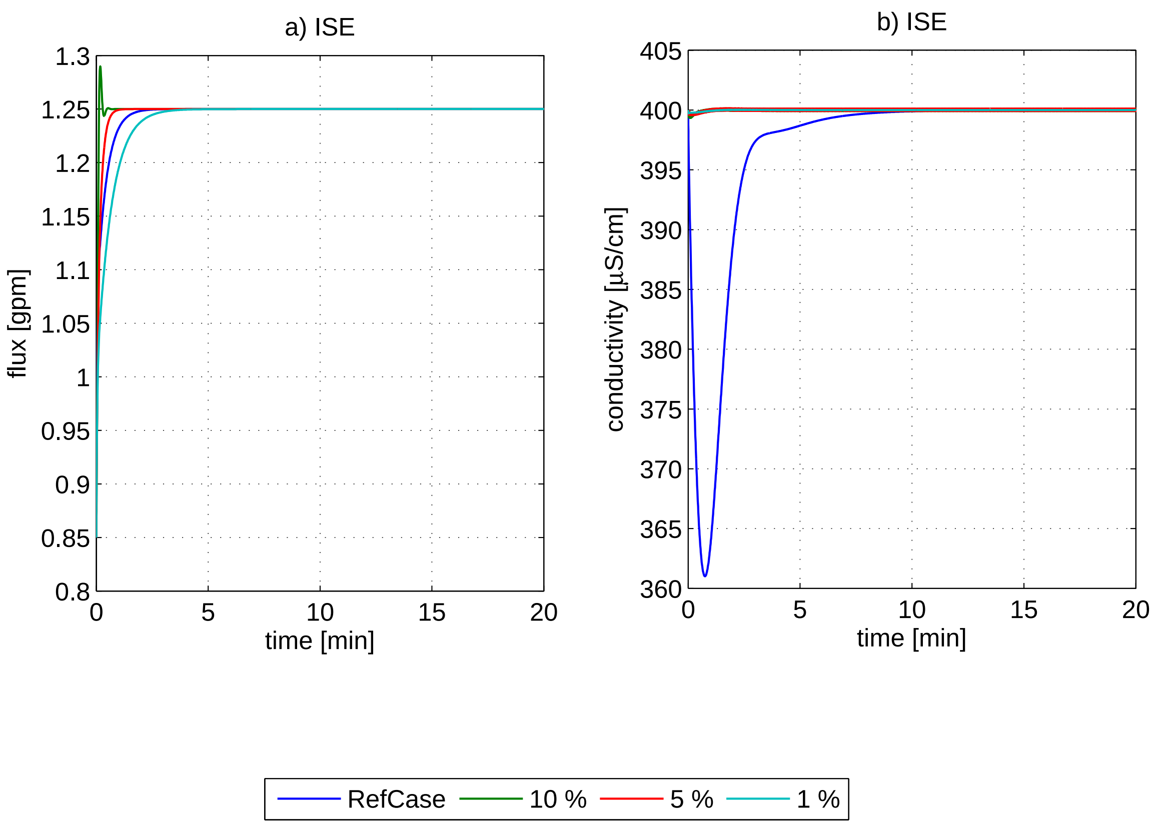

Figure 5.8. C) Output responses to a set point change of 0.4 gpm in the permeate flux according to explicit control constraints.

of the permeate conductivity are displayed in Fig. 5.9. This figure shows that all cases reach the set point within the first few seconds and independent of the accepted overshoots in the control signals.

The corresponding error signals and control signals for the ISE implementation

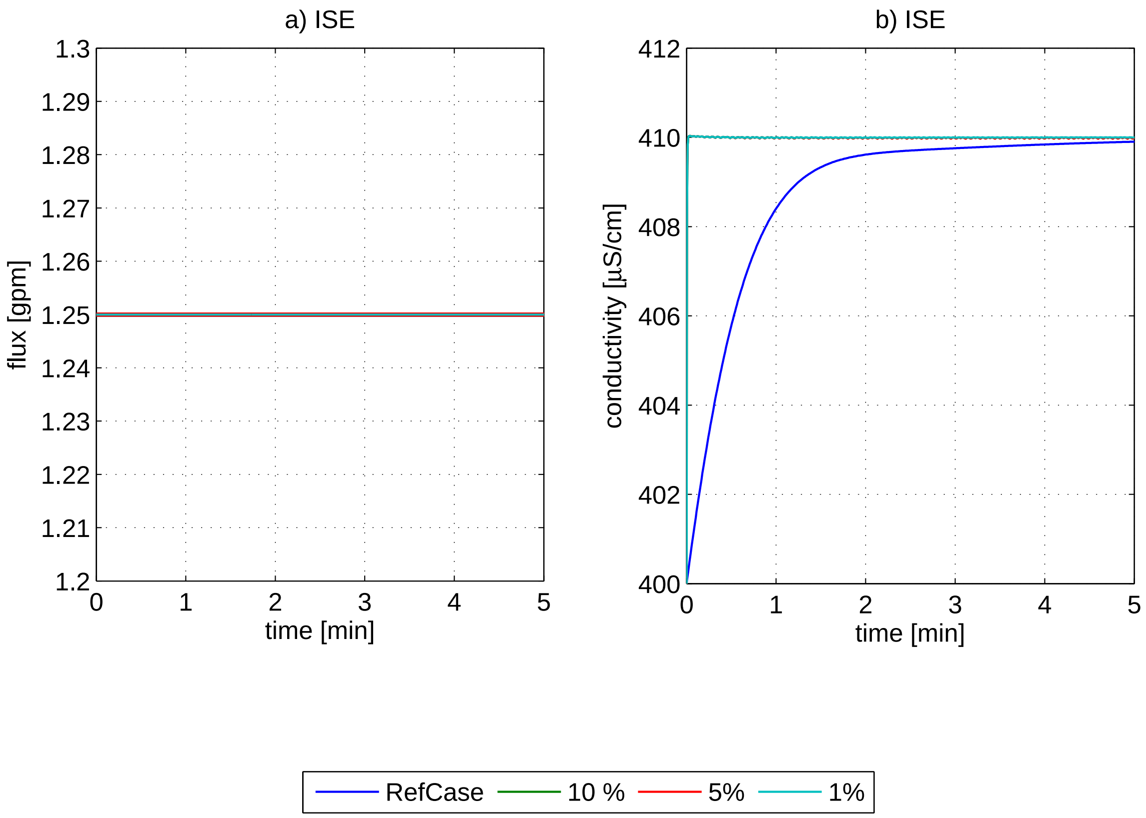

Figure 5.9. C) Output response to a set point change of 10 μS in the conductivity according to explicit control constraints.

to a step of 0.4 gpm in the permeate flux are shown in Fig. 5.10, respectively. Concerning the error signal e_1 in subplot a) of Fig. 5.10, the error convergence depends on the control constraints: the smaller the limits, the slowlier the error convergence. The control signals u_1 and u_2 in subplots c) and d) show similar behavior in relation to the explicit constraints. If less control effort is required through the constraints, the control signals need more time to reach the set point,

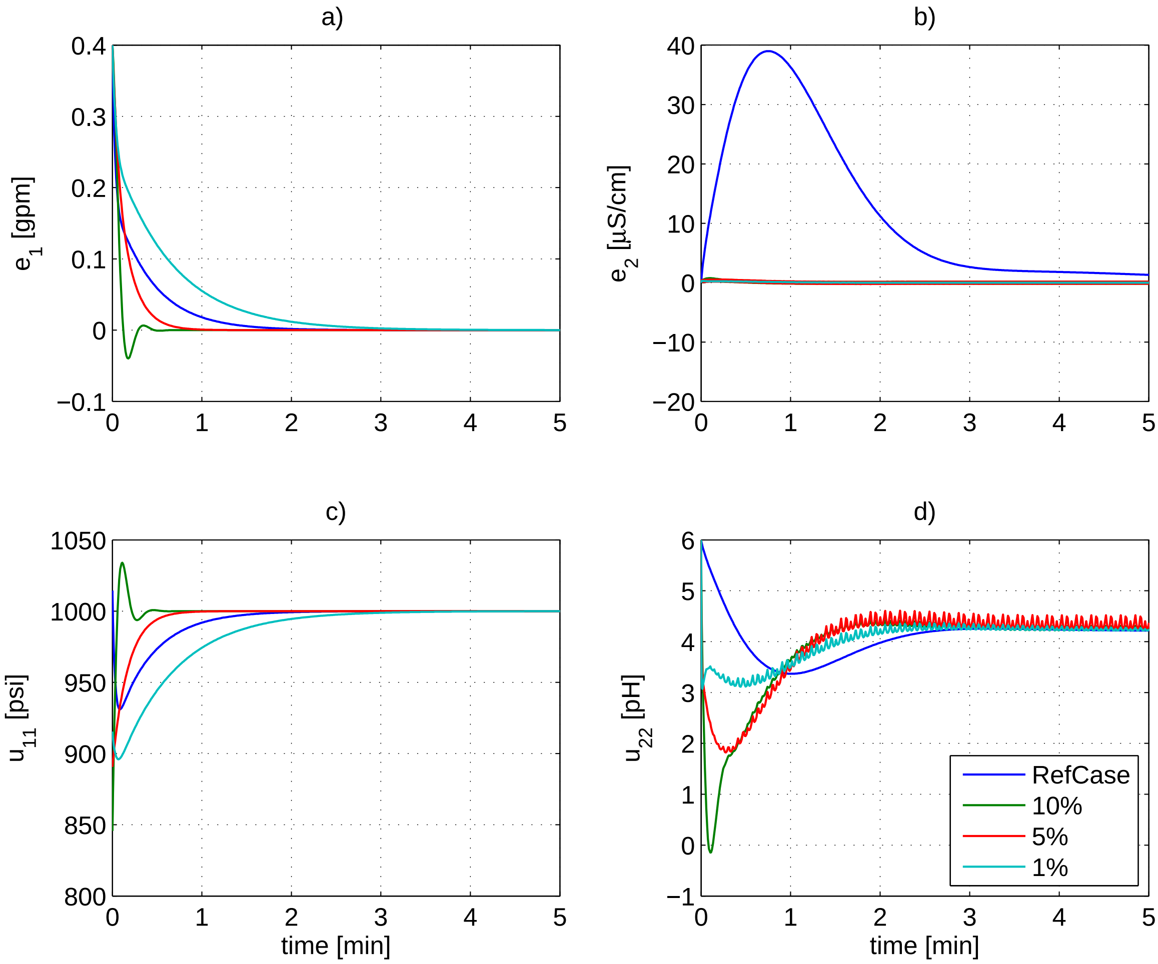

Figure 5.10. C) Responses of the error signals and control signals to a change in the flux.

but the requirement on lower control effort is met.

The presentation of the error and control signals, according to a step in the conductivity is left out in this context, due to no real new results, compared to the control effort implementations in cases A) and B).

5.2.1.5 D) Explicit control constraints as cost function implementation

For the set up of cost functions considering explicit constraints, the genetic algorithm settings are changed only in the number of cost functions to 4.

Obtaining the Pareto-optimal set and the final solution

Using the proposed approach, controllers are designed. The parameter vector $\boldsymbol{\chi}_D$ for the players is of the form

$$\boldsymbol{\chi}_D = [K_{P1},\ K_{I1},\ K_{P2},\ K_{I2}]\,, \tag{5.33}$$

with $K_{Ii} = K_{Pi}/K_{Ti}$, providing the four chromosomes for the GA. Even for case D), the corresponding ranges are set according to (5.30) and the ISE cost function implementations are applied. The implemented constraints of subsection C) are reformulated as cost functions, see (3.35).

The corresponding Nash bargaining solutions, providing the respective controller parameter sets, are listed in Table 5.7.

The statement about the dependence of the number of non dominated points and

Table 5.7. D) Controller parameters K_{P1}, K_{I1}, K_{P2} and K_{I2} for the continuous reverse osmosis system according to explicit control constraints, formulated as cost functions.

cf	case	K_{P1}	K_{I1}	K_{P2}	K_{I2}	$\#\ nondom$
	$(Ref.case)$	536	2330.435	−0.05	−0.028	−
ISE						
	10 %	411.8714	1467.5	−10	−10	170
	5 %	385.8663	1386.1	−9.666	−9.875	154
	1 %	409.7757	1088.3	−9.2405	−7.9119	71

the explicit constraints on the control signals is met in this case also: the lower the control effort is required, the lower number of non dominated points is obtained in the Pareto-optimal set.

Simulation results

Again, the step sizes are chosen as in section 5.2.1.2. According to a step of 0.4 gpm in the set point of the flux, the corresponding output responses are displayed in Fig. 5.11. Concerning the responses of the permeate flux, the responses of the reference case shows the fastest rise time as well as the fastest set point convergence, compared to all other cases. Once again, due to the triangular control system structure and the not suitable tuning method of the reference case, the corresponding responses concerning the conductivity shows the worst behavior with largest amplitude.

A step in the set point of 10 $\mu S/cm$ in the conductivity is shown in Fig. 5.12. In-

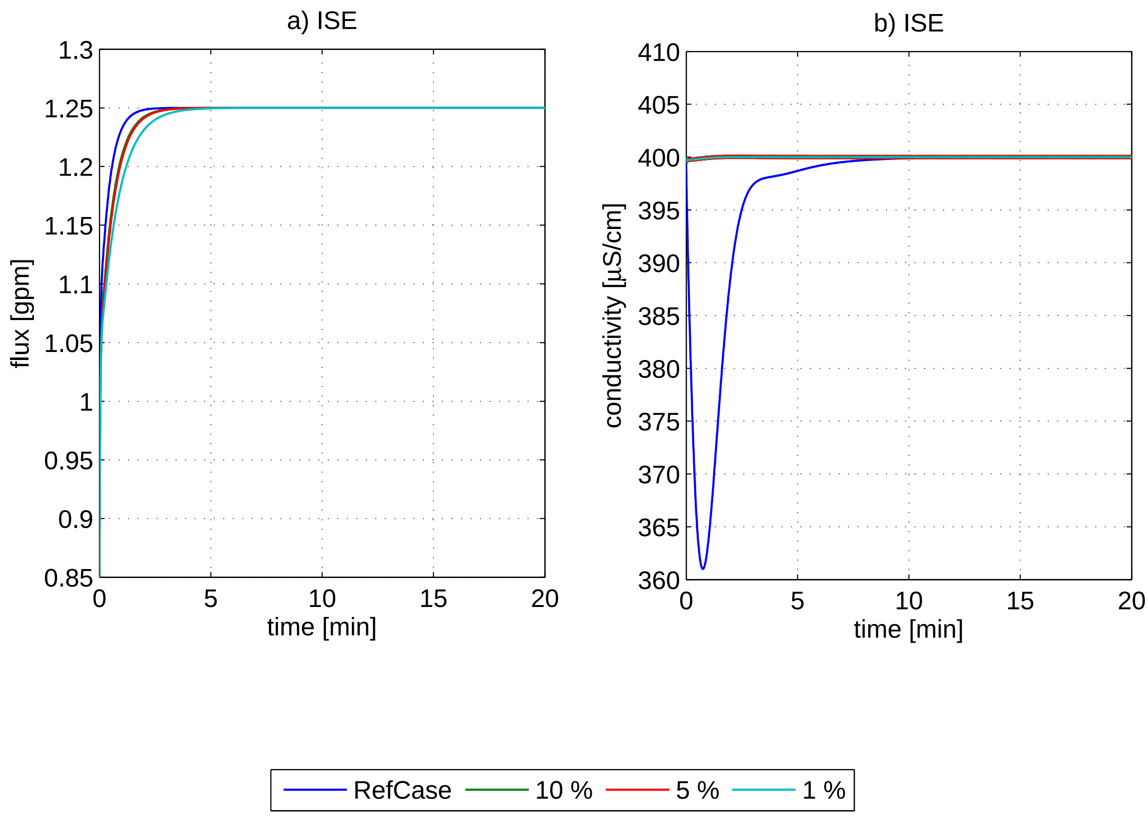

Figure 5.11. D) Response to a change in the flux according to explicit control constraints, formulated as cost functions.

dependent of the constraints, the set point in the conductivity is met very fast, compared to the reference case.

The appropriate error signals and control signals for the ISE cost function im-

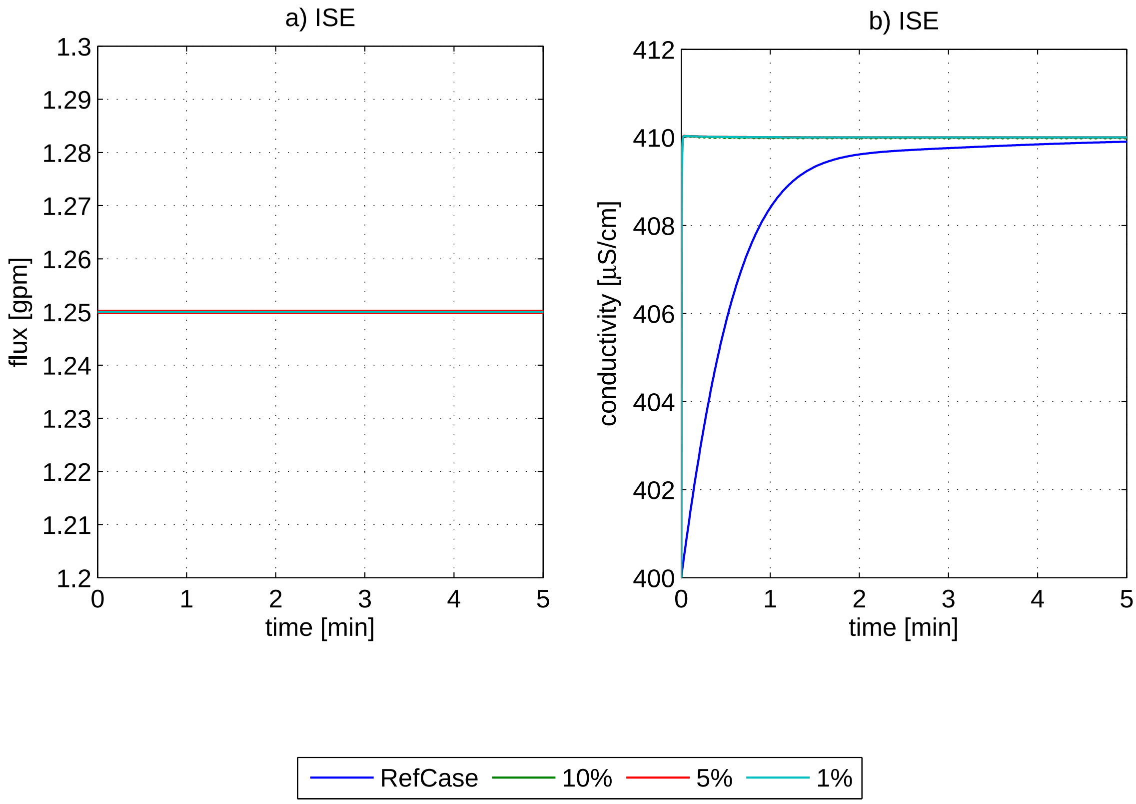

Figure 5.12. D) Response to a change in the conductivity according to explicit control constraints, formulated as cost functions.

plementation, according to a step in the permeate flux, are shown in Fig. 5.13. Equivalent to the statements concerning the outputs in Fig. 5.11, the reference case shows the fastest error convergence in e_1 and the largest error amplitude in e_2. The constraints on the control signals are kept, while several amplitudes do not differ that much as in Fig. 5.10, due to the respective cost function minimization.

The presentation of the error and control signals, according to a step in the conductivity is left out in this context, due to no real new results, compared to the control effort implementations in cases A) and B).

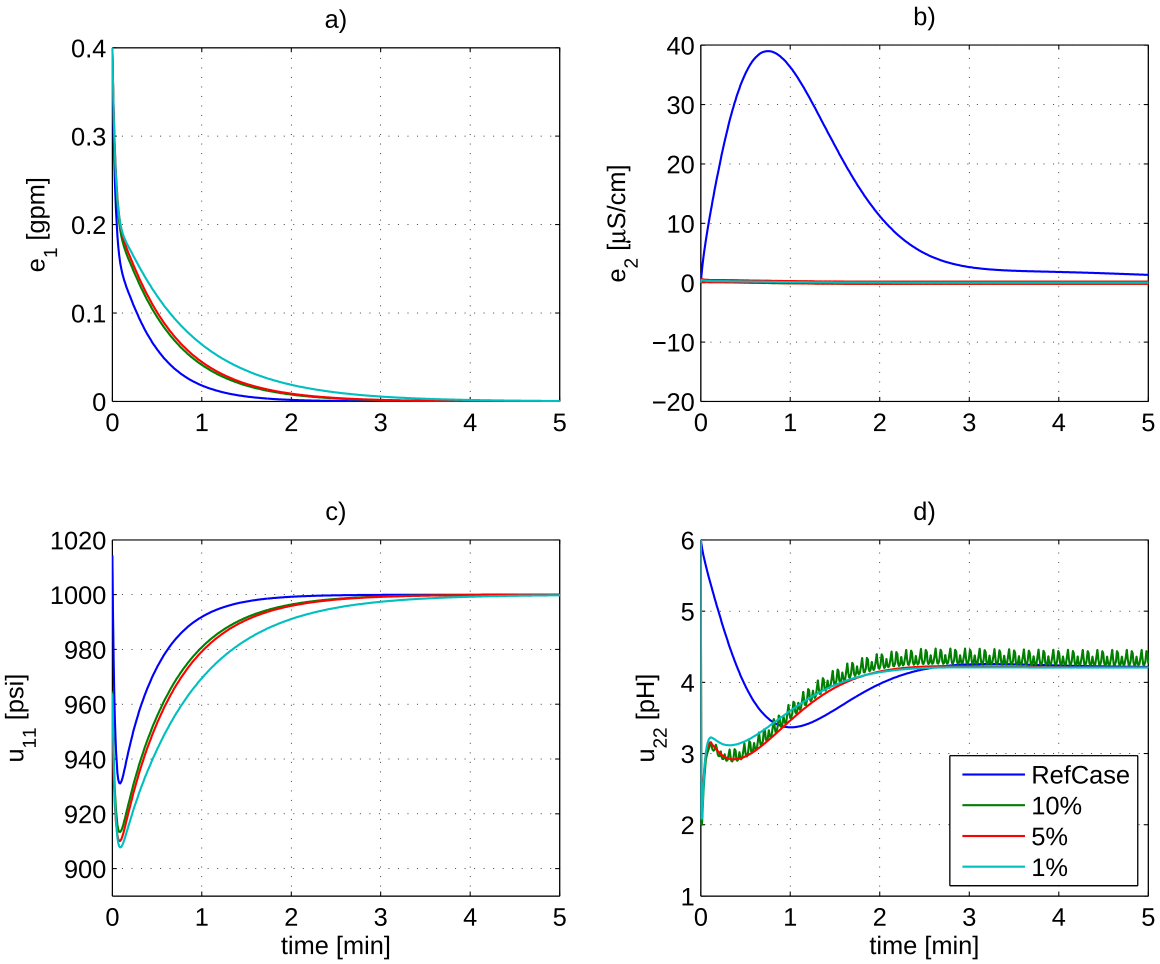

Figure 5.13. D) Response of the error signals and control signals to a change in the flux.

A performance evaluation of the different implementations on a low control effort and a conclusion related to a performance index is given in Chapter 8.

5.2.2 Multi-loop control system design including a robust stability requirement

In the second release, the specifications on the reverse osmosis system are set on a fast reference tracking with low deviation, low control effort, and robust stability of the whole system.

5.2.2.1 Cost function and constraint set up

The requirements on low control effort are realized through constraints, and maintained during the course of the game. In this part, only the ISE is transformed as a cost function for a satisfying reference tracking, with

$$J_{ec_1} = \int_0^\infty e_1^2(t)dt = \frac{1}{2\pi j} \int_{-j\infty}^{j\infty} E_1(s)E_1(-s)ds \tag{5.34}$$

and

$$J_{ec_2} = \int_0^\infty e_2^2(t)dt = \frac{1}{2\pi j} \int_{-j\infty}^{j\infty} E_2(s)E_i(-s)ds. \tag{5.35}$$

According to Subsection 5.2.1, the error signal $E_1(s)$ of the first player remain to

$$E_1(s) = \frac{A_{11}P_1}{A_{11}P_1 + B_{11}Q_1}R_1. \tag{5.36}$$

For the second player, the error signal $E_2(s)$ is

$$\begin{aligned} E_2 = &\frac{A_{22}P_2}{A_{22}P_2 + B_{22}Q_2}R_2 \\ &- \frac{B_{21}Q_1A_{11}A_{22}P_2}{A_{21}(A_{11}P_1 + B_{11}Q_1)(A_{22}P_2 + B_{22}Q_2)}R_1 \end{aligned} \tag{5.37}$$

Considering the robust stability analysis during the course of the game, modeling the control system design, with only one cost function J_μ, for the robustness requirement of the whole system is defined as

$$J_\mu = \mu(M). \tag{5.38}$$

The value of the robust stability cost J_μ, that has to be optimized, depends on the players' control strategies u_1 and u_2. Considering the cost J_μ in (5.38), with regard to the solution of the game, an additional trade-off between the robust stability and the performance of the system subject to constraints on the control effort has to be met.

The cost function J_μ, concerning the robust stability needs a computation of G_{ro}, see Fig. 5.14. The structure of G_{ro} depends on the class of uncertainty and how the uncertainties are introduced to the control structure. In this part, only parametric uncertainties are considered. For multi-loop systems, particularly MIMO systems, the consideration of parametric uncertainty is very important, since it emerges the coupling between the uncertain transfer function elements ((Skogestad u. Postlethwaite, 1996)). Thus, the parametric uncertainties are modelled as inverse additive uncertainties, compare Fig. 5.14. To distinguish between what is known and what is uncertain, the uncertainties Δ_{11}, Δ_{21}, and Δ_{22} are pulled out and placed inside a matrix block.

The computation of G_{ro}, needed for the computation of the cost function J_μ is

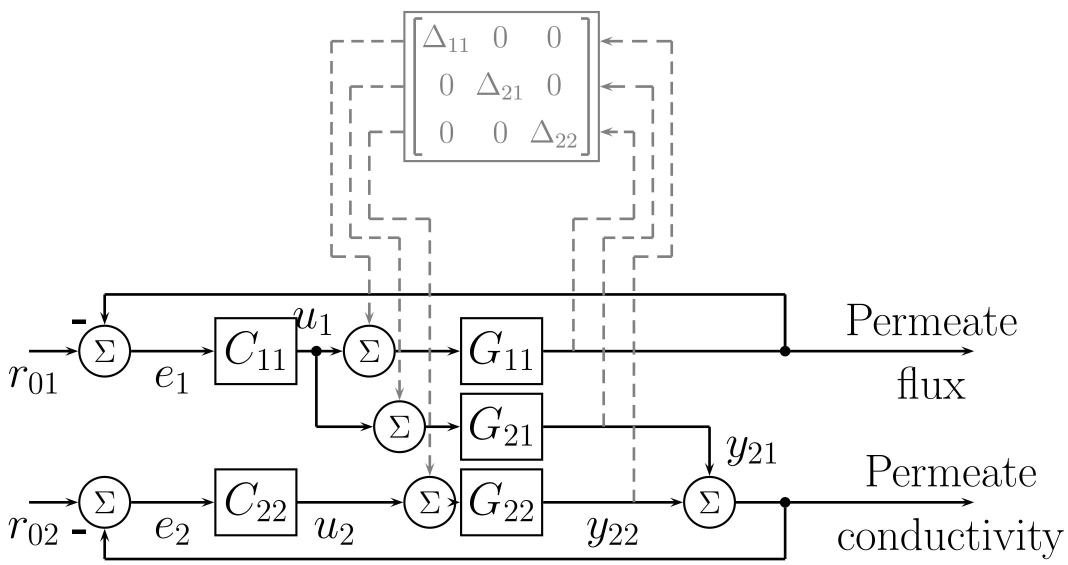

Figure 5.14. Control structure of the RO process, where the uncertain blocks Δ_{11}, Δ_{21} and Δ_{22} are pulled out and placed inside a matrix block.

done with the Matlab program *sysic*. This is a simple linear system interconnection program, writing the loop equations of the interconnections.

5.2.2.2 Obtaining the Pareto-optimal set and the final solution

The genetic algorithm operates with 200 generations and 4 chromosomes, two for each controller. Four subpopulations with 50 individuals each are chosen, and the number of cost functions is 2.

Controllers were obtained, using the GA, where the parameter vector $\boldsymbol{\chi}_\mu$ for the controllers are of the form

$$\boldsymbol{\chi}_\mu = [K_{P1},\ K_{I1},\ K_{P2},\ K_{I2}]\,, \tag{5.39}$$

with proportional K_{P1}, K_{P2} and integral $K_{I1} = K_{P1}/K_{T1}$, $K_{I2} = K_{P2}/K_{T2}$ parameters.

Obtained controller parameters are listed in Table 5.8. The parameters of games

Table 5.8. Controller and optimization parameters.

	K_{P1}	K_{I1}	K_{P2}	K_{I2}
$Game(A)$	425	10643.626	−0.48898	−0.988
$Game(B)$	501.78	11661.167	−0.071875	−0.017
$Game(C)$	450.74	3075.661	−9.156	−354.43
$Game(D)$	450.77	2905.494	−1.1444	−368.33

(A) and (B) are results of (Wellenreuther u. a., 2007), where only J_{e1}, and J_{e2} of (5.34) and (5.35) were optimized subject to predefined constraints on the control effort. In contrast, during the course of games (C) and (D), the predefined constraints on the control effort are adopted from games (A) and (B) and the cost J_μ is considered, additionally.

To be able to determine a possible relationship between constraint settings on the control effort and how robustly stable the final system is, the constraints for games (A) and (C) were chosen to be larger ($L_i = 2 \cdot u_{i_{set}}$) than those for games (B) and (D) with $L_i = 0.1 \cdot u_{i_{set}}$ subject to the $u_{i_{set}}$, corresponding control signals u_i to the set points of y_i.

5.2.2.3 Simulation Results

The operating point of the plant is given by a permeate flux of 0.85 gpm ($0.2\,{}^{m^3}\!/\mathrm{h}$) and a conductivity of $400\,{}^{\mu\,\mathrm{S}}\!/\mathrm{cm}$.

Fig. 5.15 shows the responses for the outputs (flux and conductivity) and the control signals (pressure and pH) of the nominal system for the different games $(A)-(D)$ to a change in the set point of the flux, from 0.85 gpm to 1.25 gpm, as well as a change in the set point of the conductivity from $400\,{}^{\mu S}/_{cm}$ to $430\,{}^{\mu S}/_{cm}$.

Concerning the responses of the flux (y_1), games (A) and (B) already reach the set

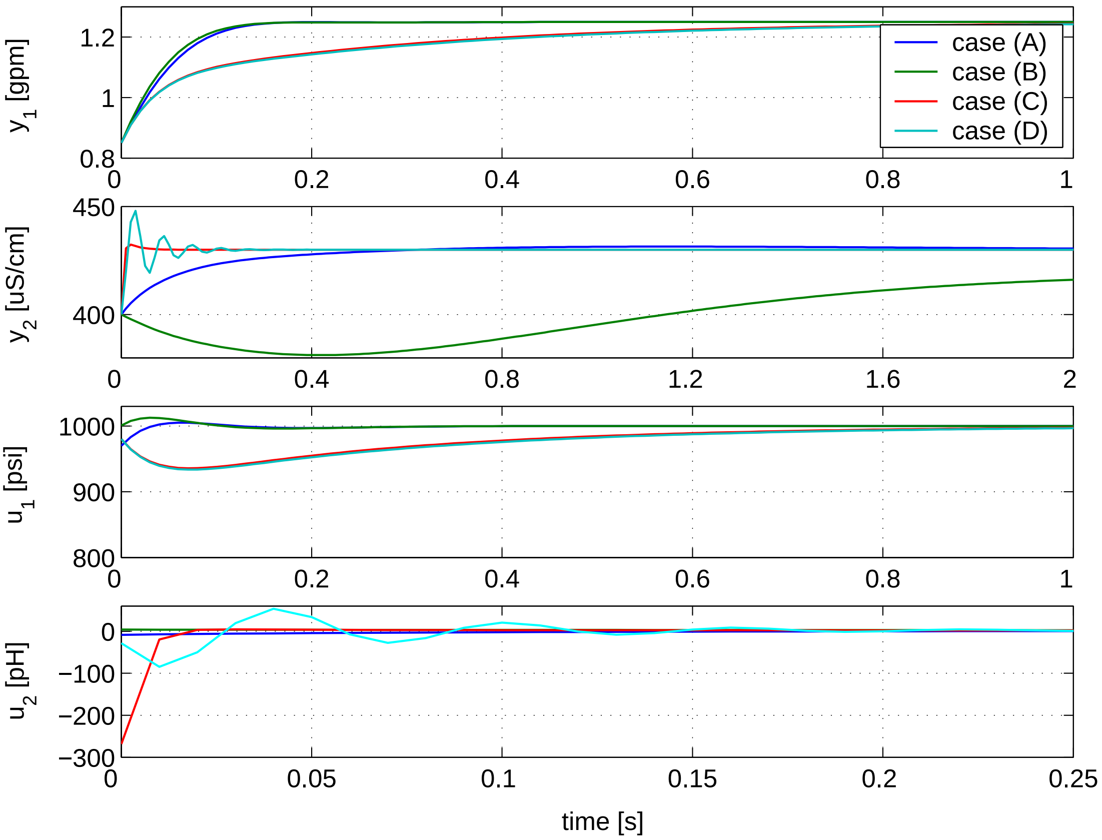

Figure 5.15. Responses to simultaneous step changes in the permeate flux y_1 and the conductivity y_2 for games $(A)-(D)$ of the nominal model.

point after 0.2 minutes, in contrast to games (C) and (D), reaching the set point not until the first minute has elapsed. All responses for the conductivity (y_2), except for game (B), reach the set point within 0.4 minutes. The control signal amplitudes for games (A) and (B) show a very similar behavior. In Fig. 5.15, concerning the control signal amplitudes of u_2 (pH), the difference between the larger constraint settings of game (C), accepting a large negative overshoot, and the narrower constraint setting of game (D) is related.

An incorporation of the robust stability consideration leads to a cost function J_μ, which is in conflict with the cost functions J_{ec_1} and J_{ec_2}. A trade-off between all three conflicting cost functions has to be found with respect to the solution of the game.

Games (A) and (B) are not robustly stable at all, compared to J_μ in Table 5.9, since this property was not considered during their optimization process. However, games (C) and (D), whose parameters are obtained with the presented approach, are robustly stable, but for different families of models, depending on the size of the structured singular value μ. For larger constraint settings (game (C)), the resulting control system is more robustly stable compared to smaller constraint settings (game (D)). The worth of the cost, concerning J_{ec_2} for game (C), degrades about 40 percent compared to game (D), while it is more robustly stable. However, the worth of the cost J_{ec_1} for game (C) improves only 5 percent compared to game (D).

According to (Skogestad u. Postlethwaite, 1996), stability is guaranteed for all

Table 5.9. Payoff function values obtained through the GA.

	J_1	J_2	J_μ
$Game(A)$	0.0180	0.5701	2.0407
$Game(B)$	0.0155	15.3980	11.3822
$Game(C)$	0.048526	0.00057632	0.51452
$Game(D)$	0.051118	0.00041084	0.82884

perturbations with appropriate structure, and $max\ \sigma\left[\Delta(j\omega)\right] \leq 1/\mu_{game}$. For the single games this yields to

$$\frac{1}{\mu_A} \approx 0.49, \quad \frac{1}{\mu B} \approx 0.088$$

and

$$\frac{1}{\mu_C} \approx 1.2065, \quad \frac{1}{\mu_D} \approx 1.943559.$$

If the admissible size of perturbation is exceeded, the stability of the system cannot be guaranteed.

5.2.2.4 Robust stability verification of the results

The RO model is changed in the domain of the different perturbation (uncertainty) sizes. This is to see which parameter sets perform better for the whole family of models, under the assumption that the perturbations are with appropriate structure. The four different perturbations are of the following size and form, where Δ is a block-diagonal structure:

$$|\Delta_1| = 0.1, \quad |\Delta_2| = 0.5$$

and

$$|\Delta_3| = 1.5, \quad |\Delta_4| = 2.0.$$

The perturbed systems are simulated according to a change in the set point of the permeate flux, and a change in the set point of the permeate conductivity, with the same sizes as the nominal system. The effects of the perturbations are shown for all games, but only for the second output y_2, the conductivity. Due to the triangular control structure, the system becomes unstable first in the lower control loop concerning the conductivity if the perturbations are too large.

Fig. 5.16 shows the step responses for all games (A)-(D). Game (B), the one with the highest cost function value concerning the robust stability, leads to an unstable closed loop system for the family of models around the nominal system and a perturbation of Δ_1. The step response of game (A) shows a larger and longer overshoot than for the nominal system, but it is still stable.

In Fig. 5.17, the representation of game (B) was neglected, since $|\Delta_1| > |\Delta_2|$, and therefore unstable in any case. Game (A) is getting unstable for a maximum perturbation of size Δ_2. The step responses of game (C) and (D) remain comparatively unchanged due to the extension of the perturbation size from Δ_1 to Δ_2 (compare Fig. 5.16 with Fig. 5.17).

An enlargement of the perturbation from Δ_2 to Δ_3 results in instability in the step responses of game (D), as shown in Fig. 5.18. Finally, Fig. 5.19 shows, that for a perturbation with structure and size of Δ_4, larger than $\frac{1}{\mu_C}$, this system is becoming unstable as well.

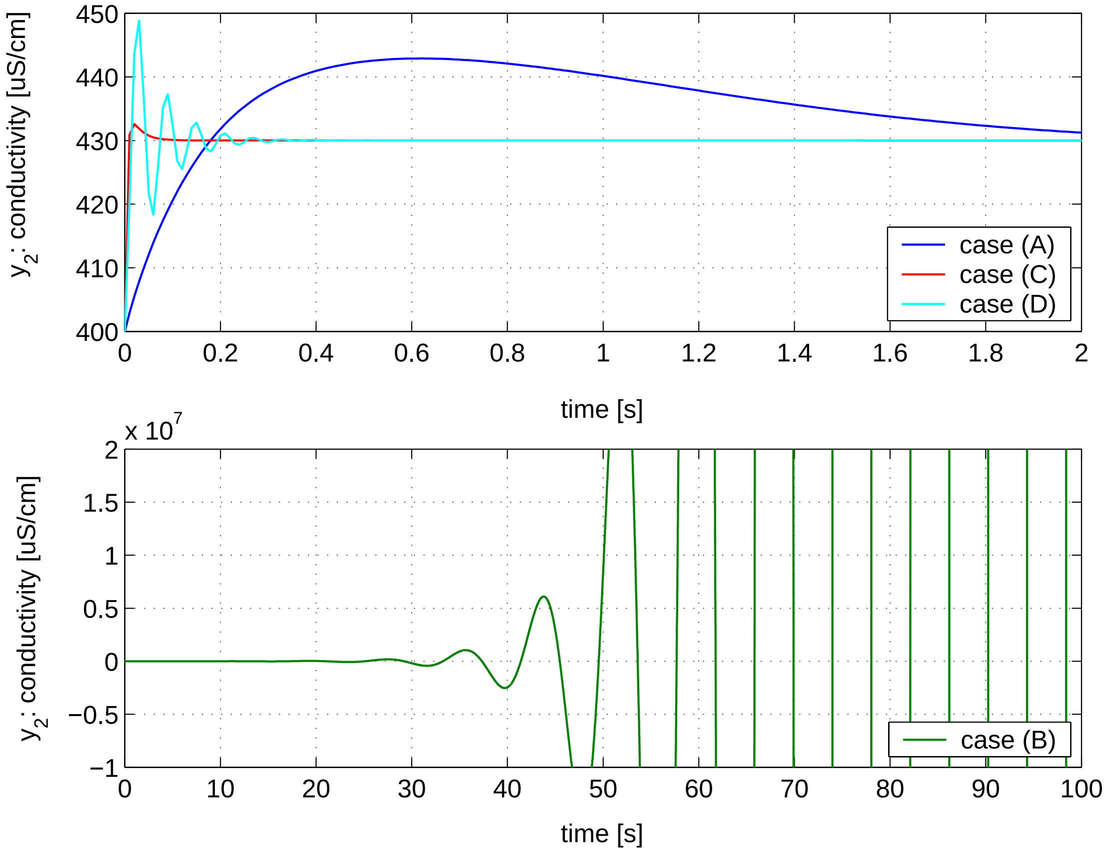

Figure 5.16. Responses to simultaneous step changes in the permeate flux y_1 and the conductivity y_2 for games $(A) - (D)$ and a perturbation of Δ_1.

Comparing all games with respect to robust performance, the robust stability indicator J_μ is smaller for all games with larger constraints than for games with smaller constraints. The system with the parameters of game (C) and the larger constraint range accepts a larger perturbation Δ before becoming more unstable than the system with the parameters of game (D).

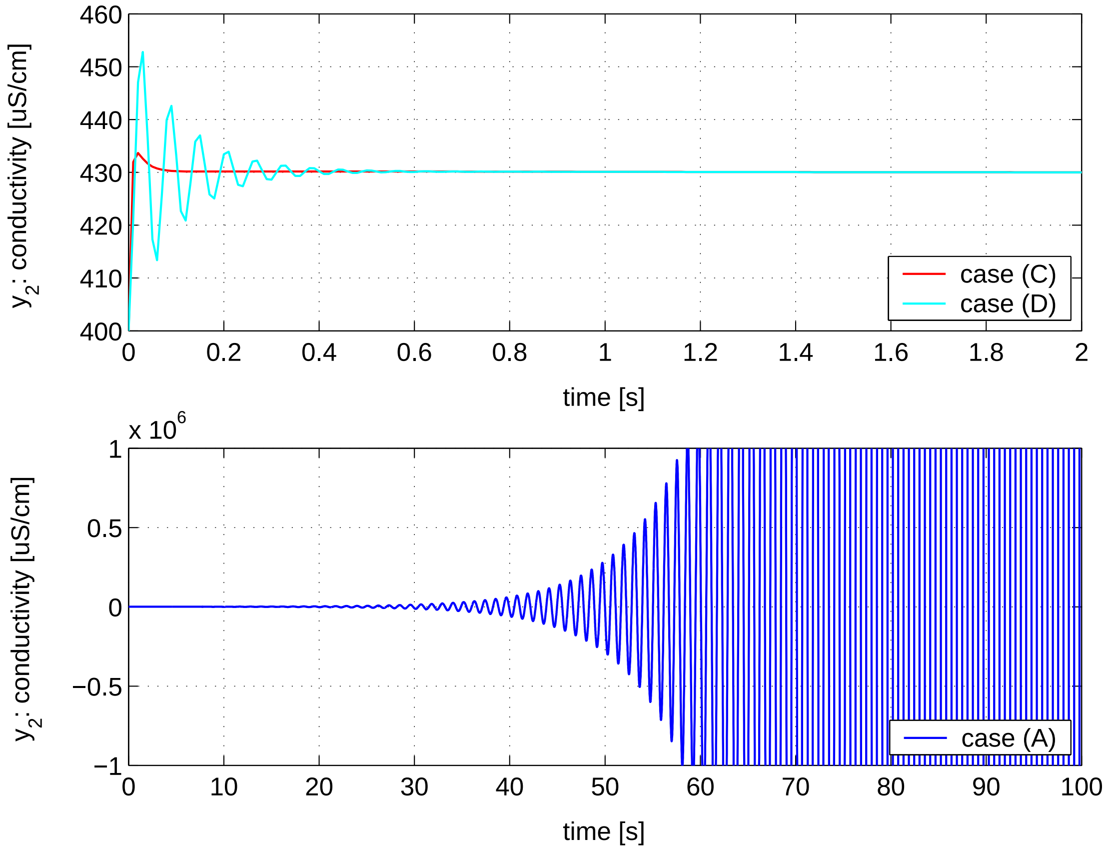

Figure 5.17. Responses to simultaneous step changes in the permeate flux y_1 and the conductivity y_2 for games (A),(C) and (D) and a perturbation of Δ_2.

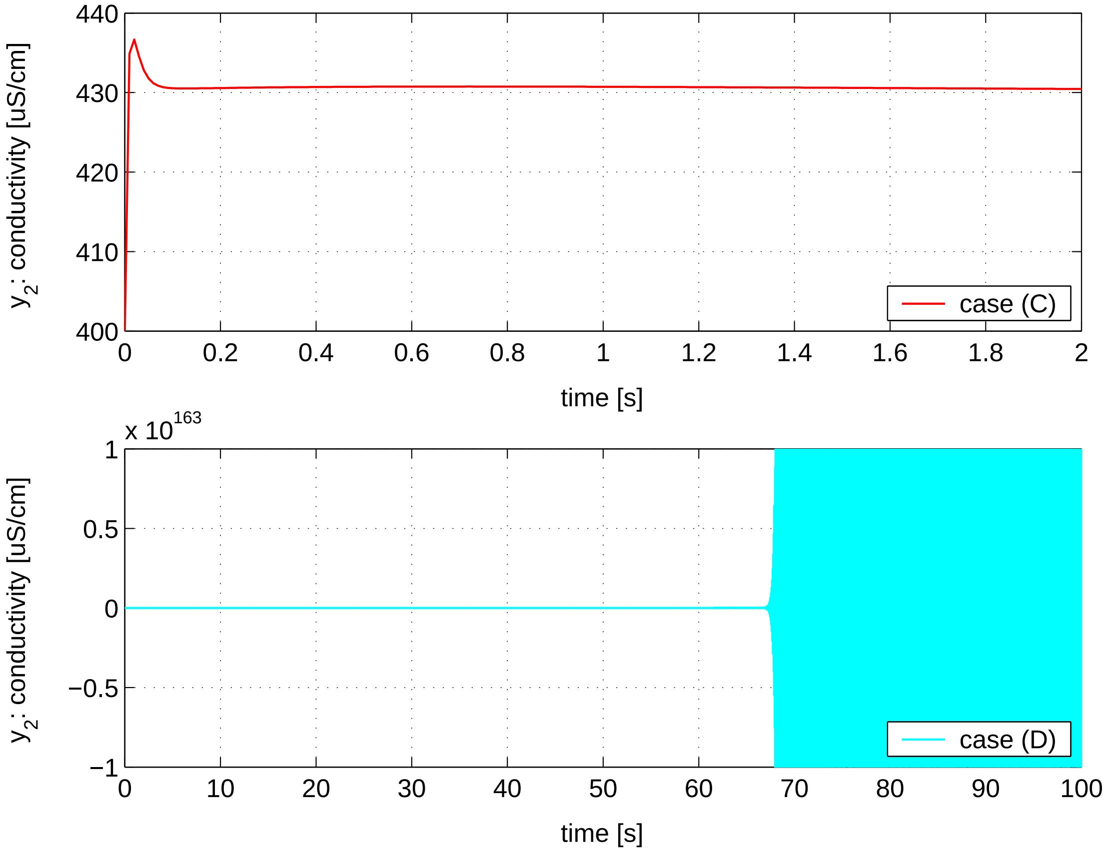

Figure 5.18. Responses to simultaneous step changes in the permeate flux y_1 and the conductivity y_2 for games (C) and (D) and a perturbation of Δ_3.

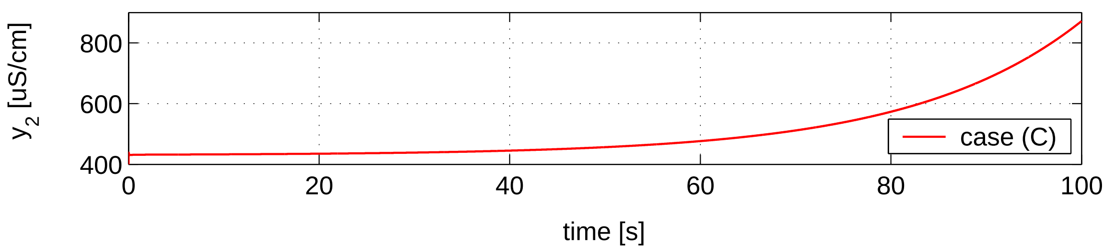

Figure 5.19. Responses to simultaneous step changes in the permeate flux y_1 and the conductivity y_2 for game (C) and a perturbation of Δ_4.

Chapter 6

Case Study 2: A Two-input/Two-output Difference Game

6.1 Multi-loop control system design for the discrete model of the reverse osmosis desalination system

The proposed game-theoretic framework is applied to the discrete representation of the reverse osmosis desalination plant. The demand on a fast reference tracking with low deviation is considered during the control system design. For the discrete game formulation, three different cost function implementations are applied and their results are compared.

6.1.1 Multi-loop control system design for a discrete plant model

The triangular control structure of a multi-loop system is shown in Fig. 5.1 in chapter 5. The controllers and the processes are described polynomial with $C_1 = Q_1/P_1$, $C_2 = Q_2/P_2$, $G_{11} = B_{11}/A_{11}$, $G_{21} = B_{21}/A_{21}$ and $G_{22} = B_{22}/A_{22}$. It is assumed that, the controllers C_1 and C_2 are PI-controllers.

6.1.1.1 Game description

The control system design with two inputs $r_{01}(k)$, $r_{02}(k)$ and two outputs $y_1(k)$ and $y_2(k)$ is considered as a dynamic difference game including two players, minimizing its own cost functions. The two controllers C_1 and C_2 are the players, each with the objective to satisfy the cost function. The strategies of the controllers are defined on

$$u_1(k) = c_1(k) * e_1(k) \tag{6.1}$$

and

$$u_2(k) = c_2(k) * e_2(k) \tag{6.2}$$

with $\mathcal{Z}\{c_1(k)\} = C_1(z) = Q_1(z)/P_1(z)$ and $\mathcal{Z}\{c_2(k)\} = C_2(z) = Q_2(z)/P_2(z)$.
The controller parameter of C_1, in Fig. 5.1, consist of Q_1 with $Q_1 = q_{1i}z + q_{0i}$. The strategies of the players are part of the strategy sets $\mathcal{U}_1 = \{u_1|u_1$ is given by (6.1)$\}$ and $\mathcal{U}_2 = \{u_2|u_2$ is given by (6.2)$\}$.
The difference game is described on the time period $[0, \infty]$ as the evolution of the errors e_1, with

$$e_1(k+6) = f_1(e_1(k+5), \ldots, e_1(k), u_1(k)), \tag{6.3}$$

and e_2, with

$$e_2(k+6) = f_2(e_2(k+5), \ldots, e_2(k), u_1(k), u_2(k)), \tag{6.4}$$

initial conditions $e_1(0) = e_{10}$ and $e_2(0) = e_{20}$, as well as the costs J_1, J_2 with $J_1 = g_{10}(e_{1\infty})$ and $J_2 = g_{20}(e_{2\infty})$. Function g_{10} is defined on $g_{10} : R_1 \times \mathcal{U}_1 \rightarrow \mathbb{R}^+$, function g_{20} is defined on $g_{20} : R_1 \times R_2 \times \mathcal{U}_1 \times \mathcal{U}_2 \rightarrow \mathbb{R}^+$.

The errors e_1, e_2 are defined on $E_i = \{e_1|e_1$ as solution of (6.3)$\}$ and $E_2 = \{e_2|e_2$ as solution of (6.4)$\}$. The function f_1 is defined on $f_1 : R_1 \times \mathcal{U}_1 \rightarrow \mathbb{R}^+$ and f_2 is defined on $f_2 : R_1 \times R_2 \times \mathcal{U}_1 \times \mathcal{U}_2 \rightarrow \mathbb{R}^+$.

6.1.1.2 Cost function and constraint set up

The stability requirement is realized through constraints, which are again kept within the course of the game. The requirements on the reference tracking is

implemented using cost functions. A satisfying reference tracking is achieved using either the ISE, the ITSE or the ISTSE cost function implementation.
For example, the cost function of the controllers J_{e_1} and J_{e_2}, concerning the error convergence, using the ISE implementation, are formulated as

$$J_{e_1} = \sum_{k=0}^{\infty} e_1^2(k) \tag{6.5}$$

and

$$J_{e_2} = \sum_{k=0}^{\infty} e_2^2(k) \tag{6.6}$$

with $e_1(k) = r_1(k) - y_1(k)$ and $e_2(k) = r_2(k) - y_2(k)$. The cost functions for the ITSE and ISTSE are given in 3.25 and 3.26.
For the controller design of the multi-loop system, the calculation of the error signals e_1 and e_2 for the costs J_{e_1} in (6.5) and J_{e_2} in (6.6) can be derived from Fig. 5.1, with the step reference signals $r_{01} = z/z-1$ and $r_{02} = z/z-1$ yielding

$$E_1 = \frac{A_{11}z}{A_{11}P_1 + B_{11}Q_1} \tag{6.7}$$

and

$$E_2 = \frac{(A_{21}A_{22}(A_{11}P_1 + B_{11}Q_1) - B_{21}Q_{11}A_{11}A_{22})z}{A_{21}(A_{11}P_1 + B_{11}Q_1)(A_{22}P_2 + B_{22}Q_2)}. \tag{6.8}$$

6.1.2 Application implementation

The proposed method is now applied on a reverse osmosis desalination plant as a discrete cooperative game, with two players and the triangular structure of Fig. 5.1. The system interaction of (5.2)-(5.5) can be rewritten for the discrete case with a sample time of $T_0 = 0.2$ as

$$G_{11}(z) = \frac{0.002013z - 2.225 \cdot 10^{-5}}{z^2 - 0.005708z + 0.001273}, \tag{6.9}$$

$$G_{21}(z) = \frac{-0.1574z + 0.08829}{z^2 - 1.383z + 0.5183} \tag{6.10}$$

and

$$G_{22}(z) = \frac{-6.084z + 3.242}{z^2 - 1.499z + 0.5488}. \tag{6.11}$$

6.1.2.1 Obtaining the Pareto-optimal set and the final solution

The two-player game between the controllers is played with the objective functions J_{e_i}. For this example, the genetic algorithm operates with 200 generations and 4 chromosomes, two for each controller. Two subpopulations with 50 individuals each are chosen, and the number of objective functions is 2. Using the proposed approach, controllers are designed, where the parameter vector $\boldsymbol{\chi}_{disc}$ for the players is of the form

$$\boldsymbol{\chi}_{disc} = [q_{11},\ q_{01},\ q_{12},\ q_{02}]\,, \tag{6.12}$$

with $Q_i = q_{1i}z + q_{0i}$, providing the four chromosomes for the GA.
The range for the parameters is set to

$$\begin{aligned} 400 &\leq q_{11} \leq 600 \\ -100 &\leq q_{01} \leq -1 \\ -0.5 &\leq q_{12} \leq -0.01 \\ 0.01 &\leq q_{02} \leq 1 \end{aligned} \tag{6.13}$$

Moreover, the controller parameters have to satisfy the constraints

$$\begin{bmatrix} -1 & 0 \\ -1 & 1 \end{bmatrix} \begin{bmatrix} q_{1i} \\ q_{0i} \end{bmatrix} < 0 \tag{6.14}$$

for the PI controller, in order to show PI behaviour. This is implicitly given through the specified parameter ranges.
Obtained controllers are given in Tab. 6.1. The number of non dominated values decrease from ISE to ITSE to the ISTSE implementation. Whereas, the values for the controller parameters are in equal ranges for all three implementations.

Table 6.1. Controller parameters q_{11}, q_{01}, q_{12} and q_{02} for the discrete reverse osmosis system.

case	q_{11}	q_{01}	q_{12}	q_{02}	*# nondom*
ISE	463.32	−3.1495	−0.31495	0.1741	42
ITSE	453.34	−2.5082	−0.33016	0.16581	37
ISTSE	478.42	−3.1656	−0.35518	0.15895	25

6.1.2.2 Simulation Results

For comparison within the three parameter sets of the proposed controller tuning method, a set point change of 0.4 gpm in the permeate flux is performed on the system, first. The corresponding step responses are shown in Fig. 6.1. The step

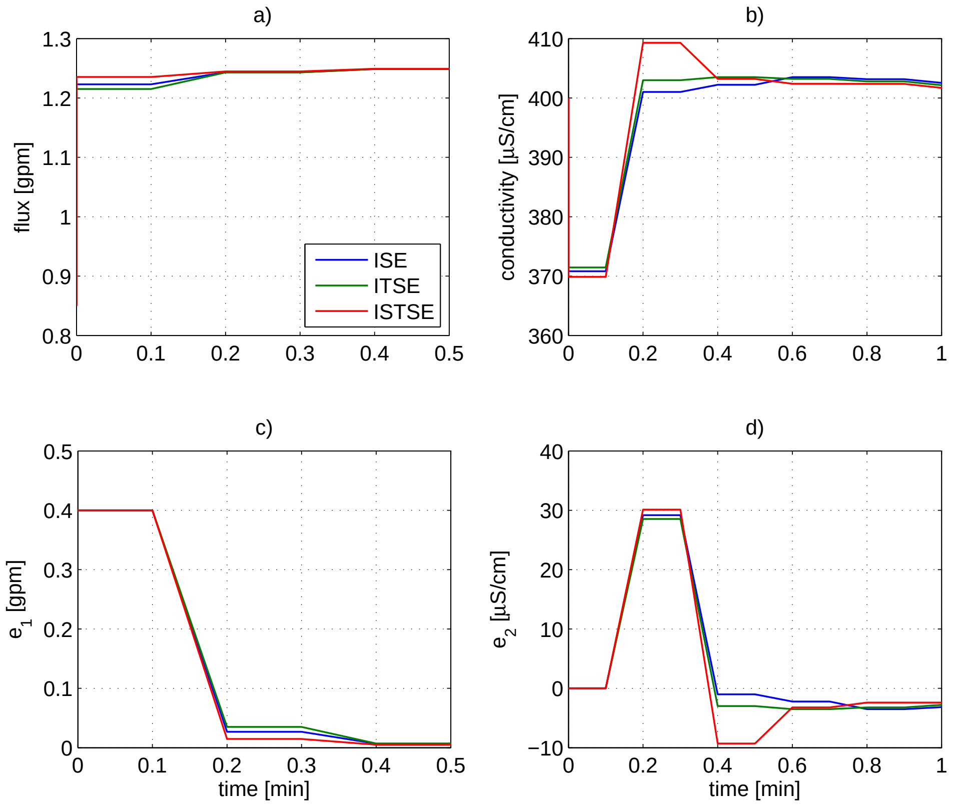

Figure 6.1. Responses of the permeate flux $y_1(k)$, the conductivity $y_2(k)$ and the errors $e_1(k)$ and $e_2(k)$ for the game-theoretic designed PI-controller according to a 0.4 gpm step in the set point of the permeate flux.

responses in subplot *a*) and *c*) for the permeate flux and it's error show very similar behaviour for all three cost function implementations. The setpoint is reached within the first minute. The resulting disturbance injection, in the set point of the conductivity due to the triangular structure, shows different step responses for the three cost function implementations in subplots *b*) and *d*). The ISTSTE cost function implementation has the largest overshoot compared to the ones for ISE and ITSE which themselves are very similar. The error signals in subplot *d*) converge to zero after approximately 2 minutes.

The responses of a 10 $\mu S/cm$ step in the set point of the conductivity is shown in Fig. 6.2. Since the interaction of the control structure is only one-way, no distur-

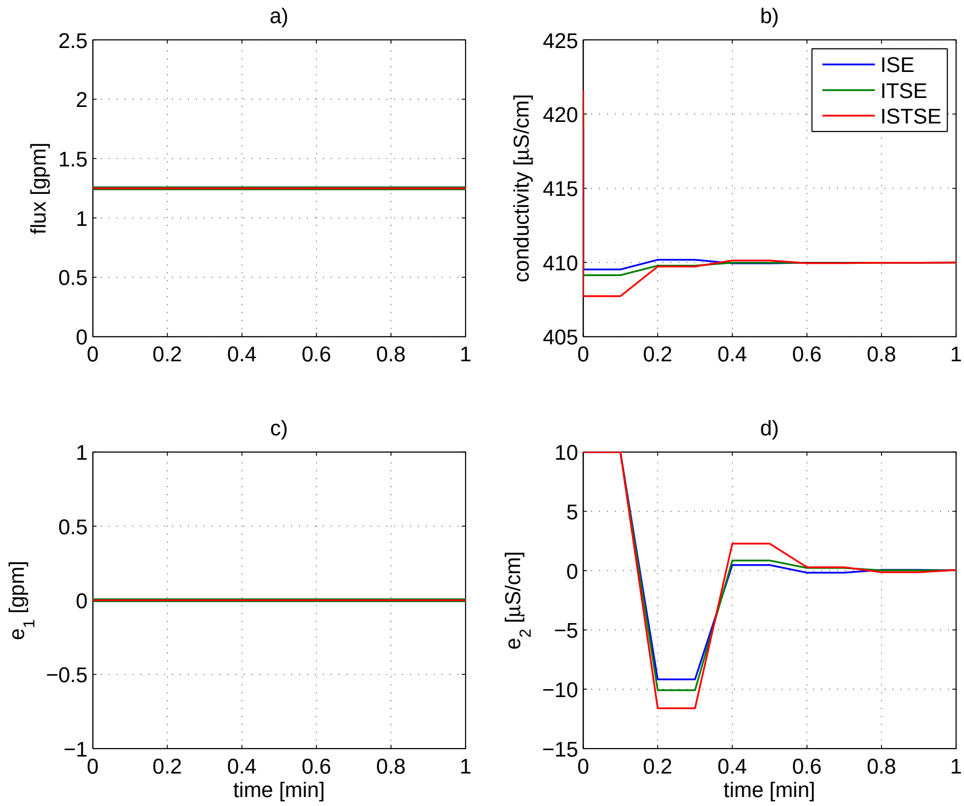

Figure 6.2. Responses of the permeate flux $y_1(k)$, the conductivity $y_2(k)$ and the errors $e_1(k)$ and $e_2(k)$ for the game-theoretic designed PI-controller according to a 10 $\mu S/cm$ step in the set point of the permeate conductivity.

bance of the set point change in the conductivity is affected in the flux. Regarding the step responses in subplots *b*) and *d*) for the conductivity and its error, the ISTSE cost function implementation shows the largest overshoot, and the step responses for the ISE and ITSE are quite similar, again. Summarizing, independent of the cost function implementation, the proposed controller tuning method provides controller parameters that show acceptable controller performance of the multi-loop control system, regarding the set point convergence. In contrast, a preference of one of the three cost function implementations could not clearly be given.

Chapter 7

Case Study 3: A Topological Analysis of different control system structures

The theory of using the developed game-theoretic framework for a topological analysis of a MIMO system is proposed in Chapter 4. In the present chapter, the topological analysis should be applied on the example of the reverse osmosis system, already used as application in Chapter 5 and Chapter 6.

7.1 Application and simulation results

To be able to compare the five different games of Section 4.2, the process transfer functions of a reverse osmosis desalination plant are used in this work with the appropriate basic control structure of Fig. 4.1 and the transfer functions of (5.2)-(5.5) from (Robertson u. a., 1996).
Five different games are implemented using the genetic algorithm (GA) of (Pohlheim, 2000). To be able to concentrate the comparisons on the different games (which is equivalent to the system structure), the requirement on the systems is focused only on a good reference tracking in the first release. The demand on a good reference tracking is implemented using the presented performance function ISE in (3.23).
An abstract of the chosen GA parameter settings are listed in Table 7.1. For games I and III, the genetic algorithm operates with 100 generations and 4 chromosomes,

Table 7.1. Algorithms and parameters for the GA.

Evolutionary algorithm	Values
Number of generations	100
Number of chromosomes	4 (2)
Subpopulations	2
Individuals (at start per subpopulation)	100,100
Number of cost functions	2 (stackelberg:1)
Selection pressure gen. gap	Stochastic universal sampling 2.1 0.9
Reinsertion rate	Local reinsertion 1
Recombination rate	Discrete recombination 1
Mutation rate range	Real valued mutation 1 0.1

two for each controller. For games II and IV, 6 chromosomes are required, due to three controllers. Two subpopulations with 100 individuals each are chosen and the number of cost functions is 2. For the stackelberg game, only 2 chromosomes are required and only one cost function is optimized in each of the two runs, one run for the leader and one run for the follower.

7.1.1 Obtaining the Pareto-optimal set and simulation results

Pareto-optimal sets as solutions of the different games are obtained, where the parameter vector $\boldsymbol{\chi}_{struct}$ of the controllers is of the form

$$\boldsymbol{\chi}_{struct} = [K_{P11}, K_{I11}, K_{P12}, K_{I12}, K_{P21}, K_{I21}, K_{P22}, K_{I22}], \tag{7.1}$$

depending on the game and providing the chromosomes for the GA.
The range of the parameters are set equally for all games. For the controller

parameters of C_{11} and C_{22} the range is set around the controller parameters of (Robertson u. a., 1996):

$$\begin{aligned} 1 &\leq K_{P11} \leq 1000 \\ 1 &\leq K_{I11} \leq 10000 \\ -100 &\leq K_{P12} \leq 100 \\ -100 &\leq K_{I12} \leq 100 \\ -100 &\leq K_{P21} \leq 100 \\ -100 &\leq K_{I21} \leq 100 \\ -10 &\leq K_{P22} \leq -0.001 \\ -10 &\leq K_{I22} \leq -0.001 \end{aligned} \tag{7.2}$$

As the controllers C_{12} and C_{21} are tuned additionally, a range for the appropriate parameters is specified where the parameters could be positive or negative with no additional constraints.

The final selection of a parameter set from the Pareto-optimal set is made using the Nash-bargaining solution concept. Obtained controller parameters, using the ISE cost function implementation, are listed in Table 7.2.

It is noticeable, that the parameter set of the stackelberg game (game V) achieves

Table 7.2. Controller and optimization parameters for the ISE cost function implementation.

	GI	GII	$GIII$	GIV	GV
K_{P11}	260.16	993.76	392.28	348.53	1000
K_{I11}	9888.7	9908.2	9996.1	9854	10000
K_{P12}	–	–	–	84.648	–
K_{I12}	–	–	–	−89.678	–
K_{P21}	–	−33.92	–	–	–
K_{I21}	–	−76.208	–	–	–
K_{P22}	−10	−9.8755	−9.9961	−10	−10
K_{I22}	−10	−3.6179	−9.9961	−10	−10
$\#nondom$	83	23	76	88	–

the predefined limits in every single parameter value. Of course, the parameter set would be different, if the limits are extended. However, the limits are chosen in the range around the controller parameters of (Robertson u. a., 1996) and should

be equal for every single game and additionally comparable to the previous case studies.
The number of non dominated points, providing the Pareto-optimal front, is comparably equal for games I, III and IV. Due to the structure, it seems to be more difficult for game II to obtain non dominated cost function pairs. This results in parameter sets keeping the system stable and satisfying the requirement on a good reference tracking.
To be able to display the dependency of the systems' outputs and the control structure derived from different information access from the game-theoretic view, the step responses of the outputs are shown and discussed in the following.
In Fig. 7.1, the step responses for the outputs y_1 and y_2 according to a change in the set point of the permeate flux from 0.85 gpm to 1.25 gpm (in SI units: from 232 m^3/h to 341 m^3/h), are shown in subplot a) and b). Considering the step responses of the flux in subplot a), all game implementations converge to the set point within 0.5 minutes, except game V (after 1 minute) and game IV (after 2.5 minutes). Game III shows the fastest set point convergence with no overshoot while the stackelberg game reaches the set point with no overshoot, as well.
The caused disturbance responses in the conductivity are shown in subplot b), where game III (with the best performance in the flux response), needs 0.5 minutes more to compensate the disturbance than the other games (which all compensate the disturbance after 3.5 minutes). The largest disturbances are caused in games I, IV and V with a size of 0.6 μS/cm.
The corresponding responses for the outputs to a change in the set point of the conductivity from 400 μS/cm to 410 μS/cm is shown in subplots c) and d). The set point step in the conductivity results only in a disturbance of +0.1 gpm and -0.03 gpm, in the flux of game IV. This is due to the control system structure, and is compensated after 0.45 minutes. In subplot d), the responses of all games show similar and fast behavior in reaching the set point of the conductivity.

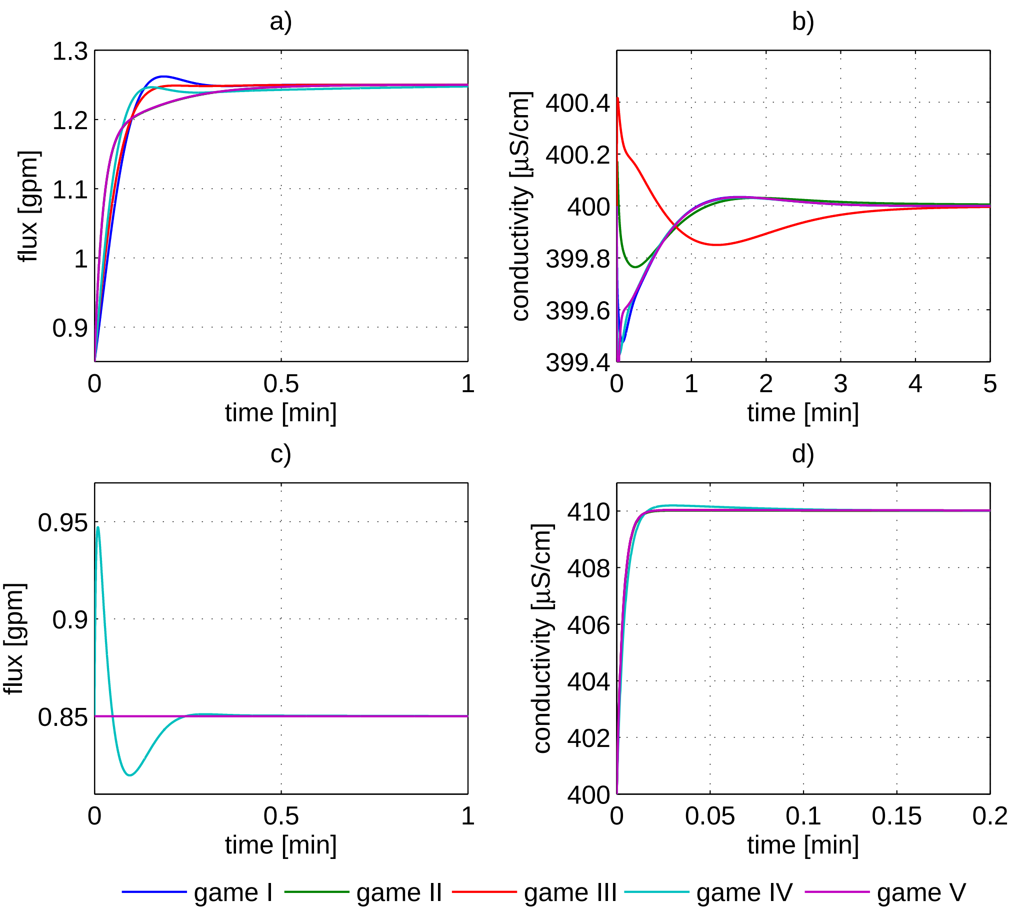

Figure 7.1. Step responses of the systems' outputs y_1, and y_2 for all games - using the ISE cost function implementation to a) and b) a step change in the set point of the permeate flux as well as c) and d) a step change in the set point of the permeate conductivity.

7.1.2 Discussion

Game I and game V are working with the same cost functions only the order of the parameter optimization is different. The amplitude responses of the figures in the previous subsection verify this difference. Game V first satisfies the cost function of the *leader*. Dependent on the result of this optimization the cost function of the *follower* is then optimized. Corresponding to this order of optimization, the step response of the flux for game V shows a faster set point convergence after a step than the one for game I. In contrast, the step response for the conductivity of game V converges more slowly to the set point compared to game I. Game I tries to find a trade-off between both cost functions while game V first satisfies

the *leader's* cost function, concerning the flux, and then the one for the *follower*, concerning the conductivity.
Game II and game III consist of the same control structure with an additional controller from e_1 to u_{22} leading to an earlier information flow. Comparing their step responses both show similar behavior. The redundant but earlier information flow from e_1 to u_{22} affects the rise time of the conductivity positively, but the step responses of the flux show a considerable overshoot of about 10% of the operating point.
Game IV, the game with the additional information of the first player, needs only a few time steps concerning the set point convergence with marginal overshoot in the step responses of the flux as well as in the step responses of the conductivity.

7.2 Additional constraints on the strategy space

In a physical system with constrained controls, the control signals are limited through a predefined size L_i:

$$|u_i| \leq L_i. \tag{7.3}$$

These constraints are maintained during the game-theoretic control system design, resulting in an optimization of (3.27).
Transferring the constraints on the control signals to game theory, they equal constraints on the strategy sets of the players.
The studied games in this section are games I, game II and game IV of Section 4.2. Game III is neglected in this consideration, because of arising problems in constrained decoupled systems (Myerson, 1991). The present study is done without game V, as well. Since, in this game, the order of optimization is more important than the structure of the system itself.
The belonging game descriptions of game I, game II and game IV are adopted from Section 4.2, additionally maintaining (7.3).

7.2.1 Comparison of structures with constrained strategy sets

The constrained strategy sets have different influence on the size and position of the Pareto-optimal sets. For a detailed study, three different limits for the control signals are set:

$$L_{1|i} = 0.01 \cdot u_{set_i} \tag{7.4}$$

$$L_{2|i} = 0.05 \cdot u_{set_i} \tag{7.5}$$

$$L_{3|i} = 0.1 \cdot u_{set_i} \tag{7.6}$$

The first limit $L_{1|i}$ for the control signals u_i tolerate a deviation on a step response of 1%, $L_{2|i}$ a deviation of 5% and $L_{3|i}$ a deviation of 10% based on u_{set_i}, corresponding to the control signals u_i in the operating points of y_i.
The Pareto-optimal sets as solutions of the proposed games, dependent on the constraints, are displayed in Fig. 7.2 for game I, in Fig. 7.3 for game II and in Fig. 7.4 for game IV.

A comparison of the Pareto-optimal sets for the players with same limit L_i show, the value of the cost function is smaller, the more deviation and the larger control signals are tolerated.
Comparing the value ranges of the cost functions, game IV has the smallest value range for both cost functions J_1 and J_2. In contrast, the value ranges for the cost functions of game I and game II take similar values for J_2, but larger values, a factor of 3, for J_1.
The Pareto-optimal front is more balanced distributed, the more Pareto-optimal points were identified, in this case: the more deviation was allowed during the optimization. In contrast, another way to obtain more Pareto-optimal points for only 5% or 1% deviation would be to increase the number of generations.
The Pareto-optimal sets for all games are obtained, using the genetic algorithm parameters of Table 5.2 with 400 cost function pairs in each iteration. The parameters of the genetic algorithm remain unchanged for every game. The number of non dominated cost function pairs, belonging to the Pareto-optimal set, varies with the size of the constraints, see Table 7.3. The narrower the limits that are chosen,

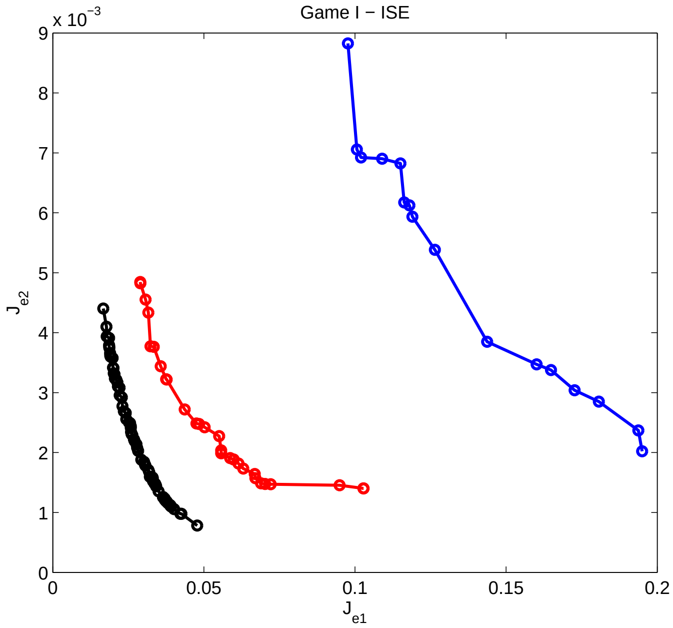

Figure 7.2. Pareto-optimal sets of game I for ISE cost function implementation with black - indexing a 10% deviation acceptance, red - indexing a 5% deviation acceptance, and blue - indexing a 1% deviation acceptance.

Table 7.3. Non dominated cost function pairs of the genetic algorithm.

	1% deviation	*5% deviation*	*10% deviation*
Game I	28	39	80
Game II	18	17	21
Game IV	7	49	68

the smaller the number of Pareto-optimal cost function pairs, obtained with the genetic algorithm, and the smaller the values of the single cost functions J_i. The number of Pareto-optimal cost function pairs also varies within the different games, influencing the distribution of the cost function pairs on the Pareto-frontier. For example, game IV, with the largest number of non dominated points, shows the Pareto-optimal set the most uniformly distributed.

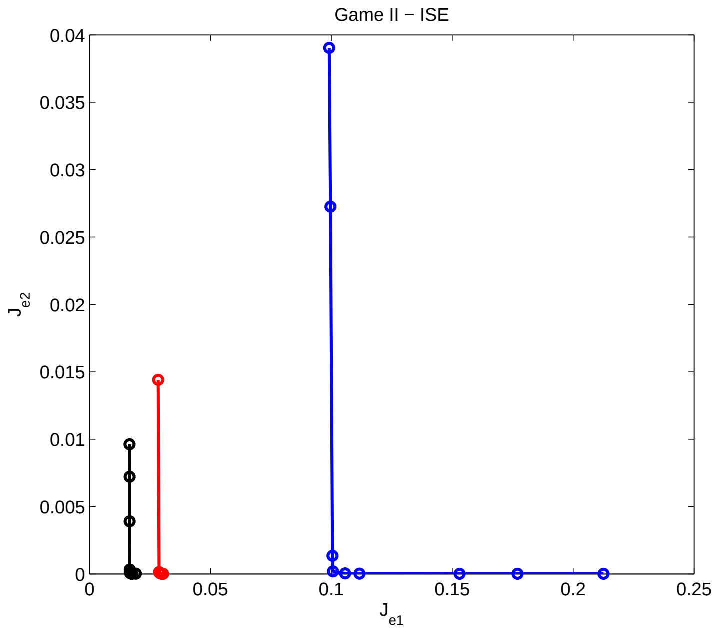

Figure 7.3. Pareto-optimal sets of game II for ISE cost function implementation with black - indexing a 10% deviation acceptance, red - indexing a 5% deviation acceptance, and blue - indexing a 1% deviation acceptance.

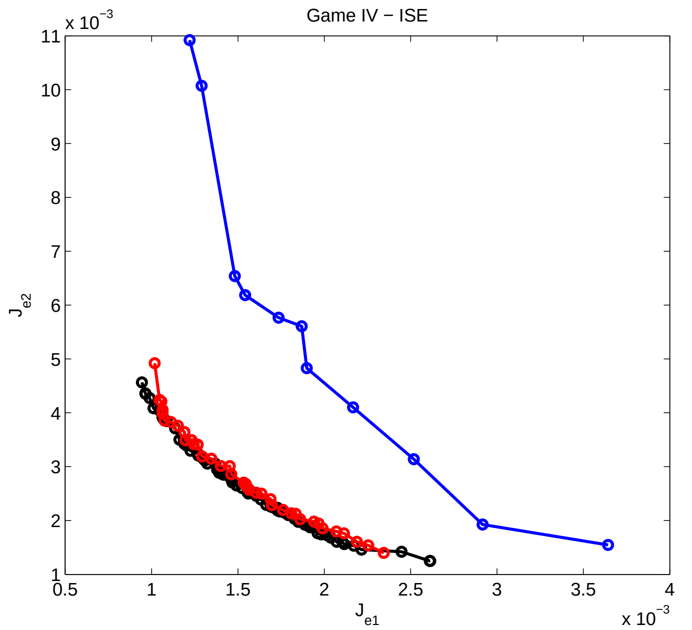

Figure 7.4. Pareto-optimal sets of game IV for ISE cost function implementation with black - indexing a 10% deviation acceptance, red - indexing a 5% deviation acceptance, and blue - indexing a 1% deviation acceptance.

Chapter 8

Evaluation of the results

In order to refine the visual comparison, based on figures of the previous chapters, performance indices are calculated during the simulation to improve the study of different cost function implementations during the control system design. Using the performance indices, refinement factors were computed relative to a reference ref. Which refinement factor and which reference is chosen for the comparison depends on which system requirements were implemented as cost functions. A refinement factor $R_i < 1$ means that the case (i) is better than the reference, ref. $R_i > 1$ means the opposite.

8.1 Evaluation of different control effort implementations

For the performance evaluation of the results of Chapter 5.2.1, performance indices of the form

$$J_{p_{e1}} = \int_0^{t_f} e_1^2(t)dt, \ J_{p_{e2}} = \int_0^{t_f} e_2^2(t)dt, \tag{8.1}$$

and

$$J_{p_{u1}} = \int_0^{t_f} u_1^2(t)dt, \ J_{p_{u2}} = \int_0^{t_f} u_2^2(t)dt. \tag{8.2}$$

are calculated during the simulation.
The corresponding refinement factors R_{e1}, R_{e2}, R_{u1} and R_{u2} are defined as

$$R_{e1} = \frac{J_{p_{e1}}}{J_{e_{1ref}}},\ R_{e2} = \frac{J_{p_{e2}}}{J_{e_{2ref}}}, \tag{8.3}$$

and

$$R_{u1} = \frac{J_{p_{u1}}}{J_{u_{1ref}}},\ R_{u2} = \frac{J_{p_{u2}}}{J_{u_{2ref}}}. \tag{8.4}$$

Upcoming, the refinement factors of the different releases are listed in Tables. The reference case in this subchapter is given through the tuned controller parameters in (Robertson u. a., 1996).

8.1.1 Evaluation of control effort added to existing cost functions (A)

In Section 5.2.1, the weighting factor λ assumes values of 0, 0.25 and 1. The performance indices are calculated during the simulation. The corresponding refinement factors are listed in Table 8.1.

Table 8.1. Refinement factors for additional weighted control effort.

Ref. factor	λ_i	RefCase	ISE
R_{e1}	0	1	0.57660
	0.25	1	42.79284
	1	1	29.05244
R_{e2}	0	1	0.00012
	0.25	1	0.00056
	1	1	0.00029
R_{u1}	0	1	0.19397
	0.25	1	80.91593
	1	1	54.50546
R_{u2}	0	1	22.99118
	0.25	1	7.98111
	1	1	7.45018

If only the error is minimized ($\hat{=} \lambda = 0$) during the optimization process (game), the ISE implementation provides a smaller refinement factor than the reference

case. In contrast, if the control effort is considered during the optimization, that means $\lambda = 0.25$ or $\lambda = 1$, additionally, this is worse than the reference case as the value is $>> 1$.
The described distribution could be recognized in the sub refinement factors R_{e1} and R_{e2} as well as R_{u1} and R_{u2}.

8.1.2 Evaluation of considering the control effort as cost function implementation (B)

If the control effort is treated as an exclusive cost function, the cost functions, concerning the error minimization as well as the cost function, minimizing the control effort are treated equally. The performance indices are calculated during the simulation, and the corresponding refinement factors are listed in Table 8.2.

Table 8.2. Refinement factors for explicit error and control effort cost functions.

Ref. factor	RefCase	ISE
R_{e1}	1	0.65346
R_{e2}	1	0.00058
R_{u1}	1	0.35237
R_{u2}	1	9.66501

Treating the control effort as cost functions, the refinement factors of the ISE implementation are always smaller than that of those for the reference case. The sub refinement factors reflect this fact, except for the factor of R_{u2}, where the reference case shows the minimum value, but with not enough strength to influence the final ranking.

8.1.3 Evaluation of applying explicit control constraints (C)

The third method of optimizing the error convergence and the control effort is to treat the control effort requirement as a constraint. Accepted deviations in the control signals are set to 1%, 5% and 10%. The appropriate refinement factors, depending on the different cost function implementations are listed in Table 8.3.

Table 8.3. Refinement factors for explicit control effort constraints.

Ref. factor	% deviation	RefCase	ISE
R_{e1}	1	1	2.38291
	5	1	1.00780
	10	1	0.64561
R_{e2}	1	1	0.00010
	5	1	0.00011
	10	1	0.00012
R_{u1}	1	1	3.40033
	5	1	0.78229
	10	1	0.34347
R_{u2}	1	1	19.14522
	5	1	21.41921
	10	1	22.88954

Starting the comparison with a 10% deviation acceptance in the control signals due to a step change, the ISE cost function implementation behave much better than the reference case, except for R_{u2}. The same applies for a 5% deviation acceptance.
However, the comparison of the refinement factors for an acceptance of 1% deviation in the control signals shows an inverted ranking. The reference case is located at the top as it does not meet any constraints. The ISE cost function implementation for 1% deviation performs worse compared to the reference case.

8.1.4 Evaluation of explicit control constraints as cost functions (D)

In the fourth release, the constraint on the control effort is extracted and treated as cost function. Again, accepted deviations in the control signals are set to 1%, 5% and 10%, and the appropriate refinement factors are listed in Table 8.4.

Compared to the approach in Subsection 8.1.3, the values for the refinements factors of the ISE cost function implementation are better now for the 1% deviation acceptance category. The refinement factors of the ISE cost function implementations are comparable for a 10% and respectively a 5% deviation acceptance in comparison with 8.1.3 but even better than the reference case.

Table 8.4. Refinement factors for explicit control effort constraints as cost functions.

Ref. factor	% deviation	RefCase	ISE
R_{e1}	1	1	2.20164
	5	1	1.81974
	10	1	1.68134
R_{e2}	1	1	0.00012
	5	1	0.00007
	10	1	0.00011
R_{u1}	1	1	3.15580
	5	1	2.40565
	10	1	2.16582
R_{u2}	1	1	18.25814
	5	1	10.57054
	10	1	20.12482

However, for a 1% deviation acceptance, the reference case provides the smallest factor, again.
The comparatively good results of the reference case in cases C) and D) are, because no constraints are kept in the reference case. Concerning the cost function implementations, there is no explicit favorite, independent of the constraint ranges.

8.1.5 Partial evaluation result concerning a fast reference tracking with low deviation and low control effort

Summarized, an explicit optimization of the cost functions for the control effort yields to better refinement factors than that of the reference case. If the control effort is optimized via a weighted sum, the refinement factors are of larger sizes compared to the cases B) to D). An equal treatment within all given system requirements, each formulated as an own cost function, leads to more balanced system responses. In case of the triangular control structure, the injected disturbance in the second control loop is not in that extent of the reference case, while keeping the control signals in specified ranges additionally.

8.2 Evaluation of the robust stability criterion

For the performance evaluation of the results of Chapter 5.2.2, performance indices of the form

$$J_{p_{e1}} = \int_0^{t_f} e_1^2(t)dt, \; J_{p_{e2}} = \int_0^{t_f} e_2^2(t)dt, \tag{8.5}$$

and

$$J_{p_{u1}} = \int_0^{t_f} u_1^2(t)dt, \; J_{p_{u2}} = \int_0^{t_f} u_2^2(t)dt, \tag{8.6}$$

are calculated during the simulation. The performance index J_μ, concerning robust stability is calculated during the game. The corresponding refinement factors R_{e1}, R_{e2}, R_{u1}, R_{u2} and R_μ are defined as

$$R_{e1} = \frac{J_{p_{e1}}}{J_{e_{1ref}}}, \; R_{e2} = \frac{J_{p_{e2}}}{J_{e_{2ref}}}, \tag{8.7}$$

$$R_{u1} = \frac{J_{p_{u1}}}{J_{u_{1ref}}}, \; R_{u2} = \frac{J_{p_{u2}}}{J_{u_{2ref}}}, \tag{8.8}$$

and

$$R_\mu = \frac{J_\mu}{J_{\mu ref}}, \tag{8.9}$$

respectively. The appropriate refinement factors for the robust stability criterion, depending on the different cost function implementations are listed in Table 8.5. The parameter set of (Robertson u. a., 1996) serves here as reverence case. Cases A) and B) are results without the consideration of robust stability but with different control effort constraints. Cases C) and D) are results, where the robust stability requirement is included in the control system design. The ranges for the control effort constraints of cases A) and C) are larger than those for cases B) and D). A distinction inside the cost function implementation is not necessary as only the ISE cost function is implemented in this release.

The refinement factors for cases A) and B), considering robust stability show maximum values, due to the disregard of the robust stability during the optimization process. However, including the robust stability criterion in the control system design, leads to low refinement values if the range for the control effort is set to a

Table 8.5. Refinement factors for an additional robust stability requirement.

Ref. factor	Case A)	Case B)	Case C)	Case D)	RefCase
R_{e1}	0.41288	0.36580	0.87223	0.92370	1
R_{e2}	0.013427	1.05118	0.00009	0.00077	1
R_{u1}	0.00703	0.00694	0.68477	0.75839	1
R_{u2}	6.87599	1.560824	22.6586	25.75321	1
R_{μ}	0.133281	0.743386	0.03360	0.05414	1

larger value, due to the minimum values in R_{μ} and R_{e2}.
The conflict between the robust stability and the size of accepted control effort is already known in the literature. This study approves of this fact. As a relationship is reproduced between the range size of the control effort and the robust stability, case A) as well as case C) provide smaller refinement factors as compared to case B) and case D), respectively.

8.2.1 Partial evaluation result concerning the reference tracking, a low control effort and robust stability

Incorporating the robust stability requirement in the control system design leads to an additional conflict between the robust stability of the system, the error minimization and the control effort constraints. Larger control effort acceptance results in a larger robust stability. This leads to a degradation of the error minimization requirement.

8.3 Evaluation of the reference tracking for the discrete game

To arrange the comparison of different cost function implementations for the requirement on a reference tracking, the performance indices and the related refinement factors are calculated as in subsection 8.1. The reference case is now given through those controller parameters, obtained through the ISE cost function implementation.

The refinement factors for the different cost function implementations, considering the control system design and according to the reference tracking, vary only

Table 8.6. Refinement factors for the discrete game subject to the reference tracking.

Ref. factor	ISE	ITSE	ISTSE
R_{e1}	1	1.0031	0.9983
R_{e2}	1	0.99	1.1715

in a small range. This is reflected by the step responses either due to a step in the flux or due to a step in the conductivity of chapter 6.

8.3.1 Partial evaluation result concerning the error minimization and stochastic disturbance compensation

As the refinement factors of the different cost function implementations vary only in a small range, an absolute implementation favorite could not be determined for this application.

8.4 Evaluation of different game structures

The influence of game theory in the topological control structure of the system is studied in Chapter 7. The corresponding performance indices and the related refinement factors are calculated as in 8.1. The references case is now given through the base triangular controller structure of Game I.

Table 8.7. Refinement factors for different game structures.

Ref. factor	Game I	Game II	Game III	Game IV	Game V
R_{e1}	1	0.50829	0.84561	0.53651	0.50393
R_{e2}	1	0.773209	0.81957	1.19825	1.10258
R_{u1}	1	2.80912	0.1604	9.1815	2.86115
R_{u2}	1	1.30322	1.16006	1.22756	1.11687

Considering the ISE cost function implementations for all games, game III is definitely on the top of the ranking, followed by game I. Game II and game V are close to each other, while the refinement factor of game IV is far away from the others due to the additional disturbing structure.

8.4.1 Partial evaluation result concerning different game structures

From control theoretic view, an additional cross connection in the control system structure as in game IV, motivated by a change in the information sets from game-theoretic view, would never be added. This could be concluded if the refinement factors for the games, using the ISE cost function implementation, are compared.

Chapter 9

Conclusions and Final Remarks

The presented approach for tuning controller parameter in multi-loop control systems is based on game theory.
The tuning of controller parameters for multiple SISO controllers in multi-loop systems results in conflicts and situations of competition due to loop interactions as well as different system requirements. The system requirements, that are considered in this work are set on stability, robust stability, reference tracking and low control effort.
The game-theoretic approach consists of a game description, including the set up of cost functions and constraints, as part of the game description. The game is described for the continuous model of a multi-loop system as a differential game. For the discrete model of a multi-loop system, the game is described as a difference game. The Pareto-optimal set represents the solution of the game. A final solution concept, applied to the Pareto-optimal solution set, provides the ultimate single solution selection.
Let's start with a game description of the multi-loop control system design. Available information of the problem is sorted and the main characteristics are identified. This includes the numbers of the players, their strategies, their outcomes or cost function values and the available information structure in the game, which is part of the rules that are established for the game. Describing the game as dynamic, with a Pareto-optimal solution set as a solution of multi-objective optimization, belongs to the rules, as well.
The set up of the cost functions, according to the corresponding system require-

ments for each player, is treated as a separate point as their development is essential. In this work, the system requirement that is set on the stability is implemented as a constraint that is kept during the optimization. The stability constraint is maintained for every application in this work, without explicitly specifying it. The demand on robust stability and a fast reference tracking with low deviation are implemented as cost functions for the continuous as well as partly for the discrete case. For the cost functions, concerning the robust stability of the system, the structured singular value is used. The cost functions considering the reference tracking of the discrete plant model, are implemented using the ISE, ITSE and ISTSE, which are well known performance functions in the literature. The system requirement with the objective of a low control effort is implemented as a constraint as well as a cost function.

After the formalization of the system requirements for the several players, all cost functions have to be optimized simultaneously. Due to the conflicts, the objective is to find a trade-off, that is accepted by each player. The problem of optimizing more than one cost function simultaneously is considered as a multi-objective optimization problem, which is solved using a genetic algorithm. The advantages of solving the Multi-objective optimization problem, using a genetic algorithm, is given through it's multiple search property and because GA's are Pareto methods, which are able to take care of all conflicting design objectives individually but compromising them concurrently. According to this, the solution of the Genetic Algorithm, that is the solution of the game, is a Pareto-optimal solution set that provides the parameter sets for the controllers. Note, that all points on the Pareto-optimal set satisfy the specified system requirements, due to the loop interactions and that a Pareto-optimal point could only increase in one cost function value, if it decreases at least one other.

As for the multi-loop control system, only one controller parameter set is needed. It is necessary to select one point of the Pareto-optimal set as final and unique solution. The final solution selection is also known as decision making and demands a solution concept, that is also derived in this work from game theory. The game-theoretic solution concept that is applied is the Nash bargaining solution concept, which has no reference to control theory. The Nash bargaining solution concept provides a solution that is fair for all players and each player is treated

equally. If a solution concept is applied, that is motivated by control theory, the Pareto-optimal solution set and the subsequent final control-theoretic solution selection can be compared to a priority based one.
In the second part of the work, the proposed approach is used for analysing the topology of a control system. This procedure results from the use of game theory, where information, or more precisely the information distribution, plays a decisive component. Dependent upon the available information distribution, that is, if it is a game with complete or incomplete information, symmetric or asymmetric, respectively, different control system topologies could arise and could be compared against each other. The advantage in using the proposed approach is given through the resulting control system topologies. These are applied not that often in common control theory.
In the last chapters of this work, the proposed approach is applied on an example of a reverse osmosis desalination plant. The reverse osmosis desalination plant is a two-input/two-output system with a triangular control structure. First, the control system of the reverse osmosis desalination plant is described as a differential game, where the requirements are set on the reference tracking and a low control effort. The implementation of the demand on low control effort is performed in four different ways, using the ISE cost function implementation and compared against each other. Formalizing a required low control effort as constraint has the advantage that, explicit ranges for the control signals could be specified. Note, it is possible that these explicit constraint ranges could be violated, as the controller parameter tuning subject to these explicit constraints is applied only to predefined step changes.
In a second release, the reverse osmosis desalination plant is described as differential game. This is subject to system requirements on the reference tracking, a low control effort and robust stability. The requirements on the reference tracking and the robust stability are expressed as cost functions, while the demand on a low control effort is formalized with explicit constraints. The simulation results as well as the performance indices indicate the conflicts of low control efforts and robust stabilities.
In the third release, the example process is described as a difference game. The requirements are set on a fast reference tracking. The developed approach is ap-

plied successfully for the given system and the belonging requirements, as well.
The application of the proposed approach as method for the setting up and evaluation of different control systems, according to one basic process, is also applied on the example of the reverse osmosis desalination system. In a first approach, five different games are set up and compared, subject to a good reference tracking. In a second approach, only three games are set up and compared, where the system requirements are set on a good reference tracking and on a low control effort, using explicit constraints.
The gain of applying game theory for the control system design problem is given through the possibility to model the mathematical relations around the strategic behavior in such situations of competition and conflict. Game theory enables a formalization that assists in the development of optimal solutions as it represents the situation of decision making explicitly.
The benefits of the proposed approach, compared to conventional methods, are given primarily in the involvement of, more or less, strong loop interactions. In addition, the proposed approach keeps the structure in such a way as no extra decoupler is necessary. In contrast, the decentral control system structure is retained as the requirements on the single control loops are treated equally and there is no composition or removal of the results. Furthermore, in obtaining the Pareto-optimal front as partial result, a relatively robust result is received as all points on the front satisfy the given system requirements.
Another benefit of the approach is given through the possibility of extending the method to other, even more complex, control system structures, as well as other possible control structures, as it's structure is not only restricted to PI or PID controllers. Finally, the game-theoretic tuned controller parameters provides an improved system behavior compared to the system behavior, achieved with conventional tuned controller parameters. Indeed, they are significantly improved, considering the simulation results and the comparison of the corresponding performance indices.
However, some critical comments remain, either through weak points in the developed approach or through still open questions.
One weak feature of the proposed algorithm is the set up of the error and control signals for the ISE, ITSE and ISTSE cost functions, which is relatively complex,

as this is still done by hand. This could be automated in that way, that an algorithm is developed which provides the relevant functions in using the control system structure. A second weakness of the developed approach is that the algorithm is computationally expensive. The reason for this is that the computations of the cost functions is performed for every individual of the population in each generation, using the evaluation algorithm of Astrom. For the present work, both of these properties are not really disadvantageous, as the approach is still an offline tuning method and there is enough time to calculate both. However, for an online application of the developed approach, these things have to be modified.
Another weakness is the use of the genetic algorithm. Due to a specified initialisation of the starting population, the obtained results differ, and often, the obtained results could not be achieved again and checked in a second optimization. Consequently, the number of generations must be large enough to produce a large Pareto-optimal set, which is uniformly distributed. Then the final solution concept selects points, that are close together if the optimization is repeated several times.
Open questions, that could be interesting for a future development are, the zero-crossing that are not considered in this work. Closing a loop around one subsystem could cause a moving of transmission zeros across the imaginary axis of other subsystems. This results in performance limitations from non minimum phase transmission zeros and is more a disadvantage of decentralized control in general, but it applies here as well. A second question that could be posed is what has to be modified if the approach should be applied to the controller tuning to a minimum phase system, which was neglected in this work, as well. Finally, the approach could be modified for the state space description of a MIMO system, and even more cost functions could be set up.

Appendix A

Further control structures

To show the usability of the developed approach to other control structures, it is now implemented first, on a cross-coupled TITO control system structure, and second, on a cascade control system structure.

A.1 The cross coupled TITO control structure

As the example structure of the thesis is only triangular, the developed approach is now applied on a TITO system with a control structure that is cross coupled, including two PI controllers, see Fig. A.1. The upper control loop interacts with

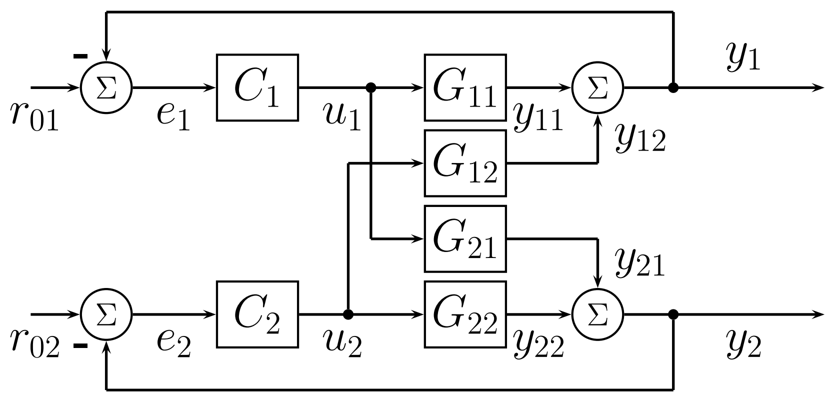

Figure A.1. Control structure of a 2x2 cross coupled process.

the lower control loop through process G_{21}, while the lower control loop interacts with the upper control loop through process G_{12}.
To tune the controller parameters of the coupled control loops of the control structure, the proposed game-theoretic approach is used for the control system design in the continuous case.

A.1.1 Application Implementation

For the game description, the process model of the distillation column is needed. According to (Skogestad u. Postlethwaite, 1996), the very crude model of a real distillation column is given as:

$$G(s) = \frac{1}{75s+1}\begin{bmatrix} 87.8 & -86.4 \\ 108.2 & -109.6 \end{bmatrix}.$$

The model of the plant is simple, but displays the important features of the distillation column behavior: it is strongly coupled.
To identify how strong the interaction of the j-th input to the i-th output is, the relative gain array (RGA) could be used (Allgöwer).
The RGA of the plant model is calculated as

$$RGA(G) = \begin{bmatrix} 35.1 & -34.1 \\ -34.1 & 35.1 \end{bmatrix}$$

The large elements in this matrix indicate that this process is fundamentally difficult to control.
Note, if the reader is interested on more, considering the RGA, he is referred to (Skogestad u. Postlethwaite, 1996). The RGA is not treated in this work before, as for a triangular control system structure because it is not significant to identify loop interactions. However, considering other, more complex, control systems, it could definitely be part of the approach. It helps in pairing the inputs and the output, if this is unknown before.

A.1.2 Multi-loop control system design

The developed game-theoretic framework, including a game description, cost function and constraint set up as well as the placement of Pareto-optimal solution set and the final solution, is applied on the idealised, simple dynamic model of a distillation column. System requirements are set only on a good reference tracking. Additionally, the application implementation and corresponding results are illustrated.

A.1.2.1 Game description

For the differential game description of the distillation column, it is assumed, that the plant is modeled by the coprime rational expressions

$$\frac{Y_{11}(s)}{U_1(s)} = G_{11}(s) = \frac{B_{11}(s)}{A_{11}(s)}, \quad \frac{Y_{12}(s)}{U_1(s)} = G_{12}(s) = \frac{B_{12}(s)}{A_{12}(s)}$$

and

$$\frac{Y_{21}(s)}{U_2(s)} = G_{21}(s) = \frac{B_{21}(s)}{A_{21}(s)}, \quad \frac{Y_{22}(s)}{U_2(s)} = G_{22}(s) = \frac{B_{22}(s)}{A_{22}(s)}.$$

The control laws of both control loops are given by

$$U_1(s) = C_1(s)E_1(s) = \frac{Q_1(s)}{P_1(s)}E_1(s) \text{ and } U_2(s) = C_2(s)E_2(s) = \frac{Q_2(s)}{P_2(s)}E_2(s).$$

The polynomial descriptions of the PI controllers C_1 and C_2 with proportional parameters K_{P_1}, K_{P_2} and integral parameters K_{T_1}, K_{T_2} are

$$C_1 = \frac{Q_1}{P_1} = \frac{K_{P_1}s + K_{P_1}/K_{T_1}}{s} \text{ and } C_2 = \frac{Q_2}{P_2} = \frac{K_{P_2}s + K_{P_2}/K_{T_2}}{s}.$$

Transferring the dynamic differential game of Section 3.1.1 to the multi-loop control system design, it is described as a differential game between two players on the time period $[0, \infty]$. The strategies of the players are defined as

$$u_1(t) = \int_0^{\infty} c_1(t)e_1(t-\tau)d\tau \tag{A.1}$$

and

$$u_2(t) = \int_0^\infty c_2(t)e_2(t-\tau)d\tau \tag{A.2}$$

with

$$\mathcal{L}\{c_1(t)\} = C_1(s) = Q_1(s)/P_1(s) \text{ and } \mathcal{L}\{c_2(t)\} = C_2(s) = Q_2(s)/P_2(s).$$

Note, that only Q_1 and Q_2 are the controller parameters of the players C_1 and C_2, as $P_1 = P_2 = 1/s$, because of the structure of the PI-controllers. The strategies of the players belong to the strategy sets $U_1 = \{u_1 | u_1 \text{ is given by } (A.7)\}$ and $U_2 = \{u_2 | u_2 \text{ is given by } (A.8)\}$.
The differential game can now be described as the evolution of the errors e_1, with

$$e_1^{(7)}(t) = f_1(e_1^{(6)}(t), e_1^{(5)}(t), e_1^{(4)}(t), e_1^{(3)}(t), \ddot{e}_1(t), \dot{e}_1(t), u_1(t), u_2(t)), \tag{A.3}$$

e_2, with

$$e_2^{(7)}(t) = f_2(e_2^{(6)}(t), e_2^{(5)}(t), e_2^{(4)}(t), e_2^{(3)}(t), \ddot{e}_2(t), \dot{e}_2(t), u_1(t), u_2(t)), \tag{A.4}$$

and initial conditions

$$e_1(0) = e_{10} \text{ and } e_2(0) = e_{20}$$

as well as a cost function for each player with

$$J_1 = g_{10}(e_{1\infty}) \text{ and } J_2 = g_{20}(e_{2\infty}).$$

The errors e_1 and e_2 belong to the sets $E_1 = \{e_1 | e_1 \text{ as solution of} (A.3)\}$ and $E_2 = \{e_2 | e_2 \text{ as solution of} (A.4)\}$, respectively. Function f_1 is defined on $f_1 : R_1 \times R_2 \times U_1 \times U_2 \to \mathbb{R}^+$ and f_2 on $f_2 : R_1 \times R_2 \times U_1 \times U_2 \to \mathbb{R}^+$ and function g_{10} on $g_{10} : R_1 \times R_2 \times U_1 \times U_2 \to \mathbb{R}^+$ as well as g_{20} on $g_{20} : R_1 \times R_2 \times U_1 \times U_2 \to \mathbb{R}^+$.

A.1.2.2 Cost function and Constraint set up

The requirements on both control loops are set on a good reference tracking. The stability is transferred to a constraint, while the requirements on the reference tracking are implemented using the ISE.

For the set up of the cost functions J_1 and J_2, according to Fig.A.1, the error signals $E_i(s)$ and the control signals $U_i(s)$, with $i = 1, 2$ of the external loop as well as the internal control are:

$$E_1 = \frac{A_{11} \cdot P_{11} \cdot A_{12} \cdot A_{21} \cdot H_2}{N} \cdot R_1 - \frac{B_{12} \cdot Q_{22} \cdot A_{11} \cdot P_{11} \cdot A_{22} \cdot A_{21}}{N} \cdot R_2,$$

$$E_2 = \frac{A_{22} \cdot P_{22} \cdot A_{21} \cdot A_{12} \cdot H_1}{N} \cdot R_2 - \frac{B_{21} \cdot Q_{11} \cdot A_{22} \cdot P_{22} \cdot A_{11} \cdot A_{12}}{N} \cdot R_1$$

and

$$U_1 = \frac{A_{11} \cdot A_{12} \cdot A_{21} \cdot H_2 \cdot Q_{11}}{N} \cdot R_1 - \frac{B_{12} \cdot Q_{22} \cdot A_{11} \cdot A_{22} \cdot A_{21} \cdot Q_{11}}{N} \cdot R_2,$$

$$U_2 = \frac{A_{22} \cdot A_{21} \cdot A_{12} \cdot H_1 \cdot Q_{22}}{N} \cdot R_2 - \frac{B_{21} \cdot Q_{11} \cdot A_{22} \cdot A_{11} \cdot A_{12} \cdot Q_{22}}{} \cdot R_1$$

with

$$H_1 = A_{11} \cdot P_{11} + B_{11} \cdot Q_{11},$$

$$H_2 = A_{22} \cdot P_{22} + B_{22} \cdot Q_{22},$$

and

$$N = (H_1 \cdot H_2 \cdot A_{12} \cdot A_{21}) - (B_{12} \cdot B_{21} \cdot Q_{11} \cdot Q_{22} \cdot A_{11} \cdot A_{22}).$$

A.1.2.3 Obtaining the Pareto-optimal set and the final solution

For the idealized model of a distillation column, the genetic algorithm operates with 100 generations and 4 chromosomes, two for each controller. Two subpopulations, with 100 individuals each, are chosen and the number of cost functions is 2.
Several controllers are designed using the proposed approach, where the parameter vector is defined as

$$\boldsymbol{\chi}_{DistCol} = [K_{P1}\ K_{I1}\ K_{P2}\ K_{I2}]^T$$

with $K_{I1} = K_{P1}/K_{T1}$ and $K_{I2} = K_{P2}/K_{T2}$.
The cost functions J_1 and J_2 are computed for different values of parameters λ_1 and λ_2.
As no controller parameters are provided in a reference case, a wide and general

range for eligible parameters of $\boldsymbol{\chi}_{DistCol}$ is chosen as:

$$1 \leq K_{P1} \leq 1000,$$

$$1 \leq K_{I1} \leq 1000,$$

$$-1000 \leq K_{P2} \leq -1, \text{and}$$

$$-1000 \leq K_{I2} \leq -1.$$

Controllers are designed for the ISE cost function implementation. A Pareto-optimal set is provided using the GA, where the Nash bargaining solution is chosen as the final solution. Obtained controller parameters for all cases are summarized in Table A.1. The amplification of the second controller is negative, as the belonging process model is negative, as well.

Table A.1. Controller parameters K_{P1}, K_{I1}, K_{P2} and K_{I2} for the distillation column.

K_{P1}	K_{I1}	K_{P2}	K_{I2}
839.23	771.67	−652.28	−771.25

A.1.3 Simulation Results

The model of the distillation column is implemented in Matlab/Simulink and simulations are carried out for the obtained parameter set. First, the output and error responses, according to a change in the set point of the first input, are shown in Fig. A.2. Considering the step responses of y_1 in subplot a), the set point is reached within 1.5 minutes, with only a small overshoot.

The caused disturbance in the second output y_2 is compensated within 3 minutes. The corresponding errors are given in the lower subplots.

Figure A.3 shows the output and error responses, according to a step in y_2.

Considering the left subplot, the set point is reached within 2 minutes, independent of the implemented cost function. The set point change in y_2 causes a disturbance in output y_1, which is compensated in 3 minutes, compare subplots showing output y_2 and it's error in Fig. A.3.

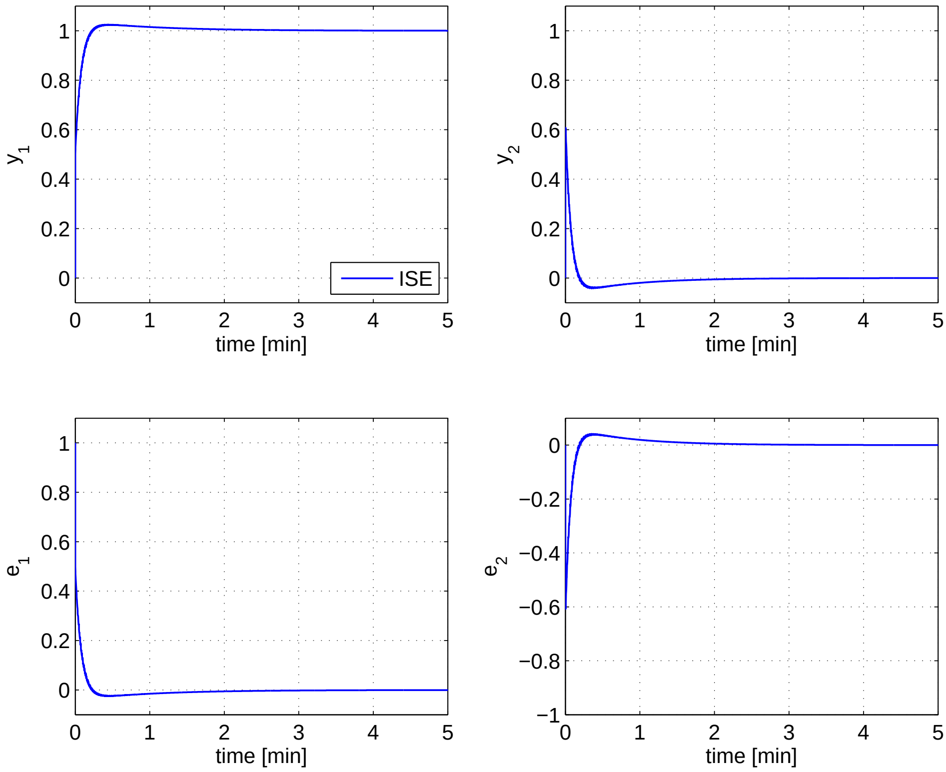

Figure A.2. Step responses of the outputs y_1 and y_2 and the errors e_1 and e_2 according to a step change in the set point of y_1.

A.1.4 Conclusion

Due to the strong interactions in the cross coupled control structure, the application of the developed game-theoretic approach shows satisfactory results in the system behavior and the mutual caused disturbances are compensated contemporary.

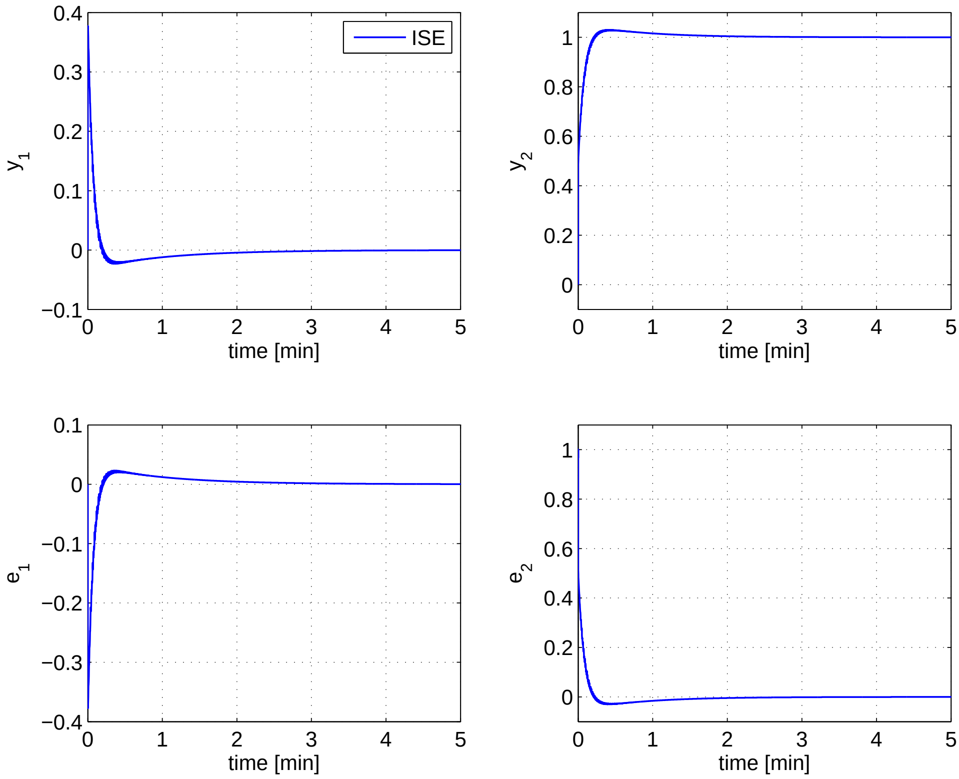

Figure A.3. Step responses of the outputs y_1 and y_2 and the errors e_1 and e_2 according to a step change in the set point of y_1.

A.2 The cascade control structure as differential game

One typical example of multi-loop control for SISO systems is the cascade structure of Fig.A.4, including two PI controllers for control intention.

A common method to tune controllers, in a cascaded control structure, is to tune them separately, starting with the most inner. In doing so, it is assumed, that the loop dynamics increases from the most inner to the last. However, this is not always the case. There are also processes with only one input but two outputs. The proposed approach in the present thesis is applied on a cascaded control structure. This is to compare the results and the application of the approach with the general

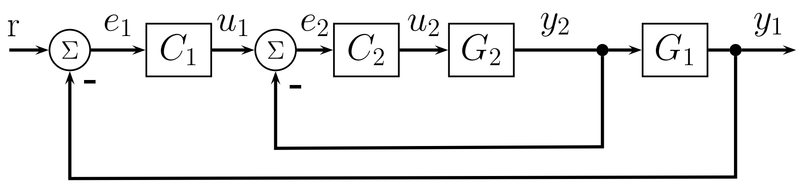

Figure A.4. Cascade structure for the heat exchanger control system.

tuning method.

To tune the controller parameters of the coupled control loops of the cascade control structure, the proposed game theoretic framework is used for the control system design in the continuous case.

A.2.1 Application Implementation

For the game description, the process model of the heat exchanger example is needed. According to (Erickson u. Hedrick, 1999), the transfer function for the steam flow process of the internal loop is a first order plus time delay model:

$$G_2(s) = \frac{1.0}{0.10s + 1} e^{-0.02s}. \tag{A.5}$$

The transfer function for the temperature process of the external loop is also a first order plus deadtime model:

$$G_1(s) = \frac{0.73}{0.65s + 1} e^{-0.19s}. \tag{A.6}$$

The second order Pade Approximations of the deadtime models, in equations (A.5) and (A.6), yield to the corresponding transfer functions:

$$\frac{B_1(s)}{A_1(s)} = G_1(s) = \frac{0.73s^2 - 23.0526s + 242.6593}{0.65s^3 + 21.5263s^2 + 247.6454s + 332.41}$$

and

$$\frac{B_2(s)}{A_2(s)} = G_2(s) = \frac{s^2 - 300s + 30000}{0.1s^3 - 31s^2 + 3300s + 30000}.$$

A.2.2 Multi-loop control system design

The developed game-theoretic framework, including a game description, cost function and constraint set up as well as the placement of Pareto-optimal solution set and the final solution, is applied on the continuous heat exchanger system in detail. System requirements are set on a good reference tracking as well as a low control effort. Additionally, the application implementation and corresponding results are illustrated.

A.2.2.1 Game description

For the differential game description of the heat exchanger, it is assumed that, the plant is modeled by the coprime rational expressions

$$\frac{Y_1(s)}{U_1(s)} = G_1(s) = \frac{B_1(s)}{A_1(s)} \text{ and } \frac{Y_2(s)}{U_2(s)} = G_2(s) = \frac{B_2(s)}{A_2(s)}.$$

The control laws of both control loops are given by

$$U_1(s) = C_1(s)E_1(s) = \frac{Q_1(s)}{P_1(s)}E_1(s) \text{ and } U_2(s) = C_2(s)E_2(s) = \frac{Q_2(s)}{P_2(s)}E_2(s).$$

The polynomial descriptions of the PI controllers C_1 and C_2 with proportional parameters K_{P_1}, K_{P_2} and integral parameters K_{T_1}, K_{T_2} are

$$C_1 = \frac{Q_1}{P_1} = \frac{K_{P_1}s + K_{P_1}/K_{T_1}}{s} \text{ and } C_2 = \frac{Q_2}{P_2} = \frac{K_{P_2}s + K_{P_2}/K_{T_2}}{s}.$$

Transferring the dynamic, differential game of Section 3.1.1 to the multi-loop control system design, it is described as a differential game between two players on the time period $[0, \infty]$. The strategies of the players are defined as

$$u_1(t) = \int_0^\infty c_1(t)e_1(t-\tau)d\tau \tag{A.7}$$

and

$$u_2(t) = \int_0^\infty c_2(t)e_2(t-\tau)d\tau \tag{A.8}$$

with

$$\mathcal{L}\left\{c_1(t)\right\} = C_1(s) = Q_1(s)/P_1(s) \text{ and } \mathcal{L}\left\{c_2(t)\right\} = C_2(s) = Q_2(s)/P_2(s).$$

Note, only Q_1 and Q_2 are the controller parameters of the players C_1 and C_2, as $P_1 = P_2 = 1/s$, due to structure of the PI-controllers. The strategies of the players belong to the strategy sets $U_1 = \{u_1 | u_1 \; is \; given \; by \; (A.7)\}$ and $U_2 = \{u_2 | u_2 \; is \; given \; by \; (A.8)\}$.

The differential game can now be described as the evolution of the errors e_1, with

$$e_1^{(6)}(t) = f_1(e_1^{(5)}(t), e_1^{(4)}(t), e_1^{(3)}(t), \ddot{e}_1(t), \dot{e}_1(t), u_1(t), u_2(t)), \tag{A.9}$$

e_2, with

$$e_2^{(6)}(t) = f_2(e_2^{(5)}(t), e_2^{(4)}(t), e_2^{(3)}(t), \ddot{e}_2(t), \dot{e}_2(t), u_1(t), u_2(t)), \tag{A.10}$$

and initial conditions

$$e_1(0) = e_{10} \text{ and } e_2(0) = e_{20}$$

as well as a cost function for each player with

$$J_1 = g_{10}(e_{1\infty}) \text{ and } J_2 = g_{20}(e_{2\infty}).$$

The errors e_1 and e_2 belong to the sets $E_1 = \{e_1 | e_1 \; as \; solution \; of (A.9)\}$ and $E_2 = \{e_2 | e_2 \; as \; solution \; of (A.10)\}$, respectively. Function f_1 is defined on $f_1 : R_1 \times U_1 \times U_2 \rightarrow \mathbb{R}^+$ and f_2 on $f_2 : R_2 \times U_1 \times U_2 \rightarrow \mathbb{R}^+$ and function g_{10} on $g_{10} : R_1 \times U_1 \times U_2 \rightarrow \mathbb{R}^+$ as well as g_{20} on $g_{20} : R_2 \times U_1 \times U_2 \rightarrow \mathbb{R}^+$.

A.2.2.2 Cost function and Constraint set up

The requirements on both control loops are set on a good reference tracking and low control effort. The stability is transferred to a constraint. While, the requirements on a good reference reaction and low control effort are combined in a cost function with a weighting factor λ, and implemented using the ISE cost function implementation.

For the set up of the cost functions J_1 and J_2, according to Fig.A.4, the error signals $E_i(s)$ and the control signals $U_i(s)$ of the external loop as well as the internal control loop are:

$$E_1 = \frac{(P_2A_2 + Q_2B_2)A_1P_1}{(P_2A_2 + Q_2B_2)A_1P_1 + Q_2B_2B_1Q_1}R,$$

$$E_2 = \frac{A_1A_2Q_1P_2}{(P_2A_2 + Q_2B_2)A_1P_1 + Q_2B_2B_1Q_1}R.$$

as well as

$$U_1 = \frac{(P_2A_2 + Q_2B_2)A_1Q_1}{(P_2A_2 + Q_2B_2)A_1P_1 + Q_2B_2B_1Q_1}R,$$

$$U_2 = \frac{A_1A_2Q_1Q_2}{(P_2A_2 + Q_2B_2)A_1P_1 + Q_2B_2B_1Q_1}R,$$

A.2.2.3 Obtaining the Pareto-optimal set and the final solution

For the continuous heat exchanger application, the genetic algorithm operates with 100 generations and 4 chromosomes, two for each controller. Two subpopulations, with 100 individuals each, are chosen, and the number of cost functions is 2.
For the example of the heat exchanger, several controllers are designed using the proposed approach, where the parameter vector is defined as

$$\boldsymbol{\chi}_{HeEx} = [K_{P1}\ K_{I1}\ K_{P2}\ K_{I2}]^T,$$

with $K_{I1} = K_{P1}/K_{T1}$ and $K_{I2} = K_{P2}/K_{T2}$.
The cost functions J_{cf_1} and J_{cf_2} are computed for different values of the weighting factors λ_1 and λ_2.
First, a range for eligible parameters of $\boldsymbol{\chi}_{HeEx}$ is chosen around the tuned parameters in (Erickson u. Hedrick, 1999):

$$0.1 \le K_{P1} \le 5,$$

$$0.1 \le K_{I1} \le 5,$$

$$0.1 \le K_{P2} \le 5,$$

$$0.1 \le K_{I2} \le 50.$$

Controllers are designed for three different values of λ, $\lambda_1 = \lambda_2 = 1$, $\lambda_1 = \lambda_2 = 0.25$, and $\lambda_1 = \lambda_2 = 0$, using the cost function implementation of the ISE. The controllers are referenced as case $\lambda_i = 1$, case $\lambda_i = 0.25$ and case $\lambda_i = 0$, respectively. A Pareto-optimal set is provided using the GA, where the Nash bargaining solution is chosen as the final solution. Obtained controller parameters for all cases are summarized in Table A.2.

Table A.2. Controller parameters K_{P1}, K_{I1}, K_{P2} and K_{I2} for the continuous heat exchanger system.

case	K_{P1}	K_{I1}	K_{P2}	K_{I2}
$\lambda_i = 1$	0.1	1.7478	5	0.97418
$\lambda_i = 0.25$	0.63833	1.9983	5	1.9983
$\lambda_i = 0$	0.97179	1.9998	4.9995	1.9998

A.2.3 Simulation Results

The model of the heat exchanger example is implemented in Matlab/Simulink and simulations are carried out for all controllers. Step responses of the outlet temperature y, according to a change of $1K$ in the set point of the outlet temperature, are shown in Fig.A.5, for each case. The comparison of the step responses, in Fig.A.5, leads to the result that, for $\lambda_i = 0$, the overshoots are the smallest. They are followed by $\lambda_i = 0.25$ and $\lambda_i = 1$. The reference case shows the slowliest set point convergence.

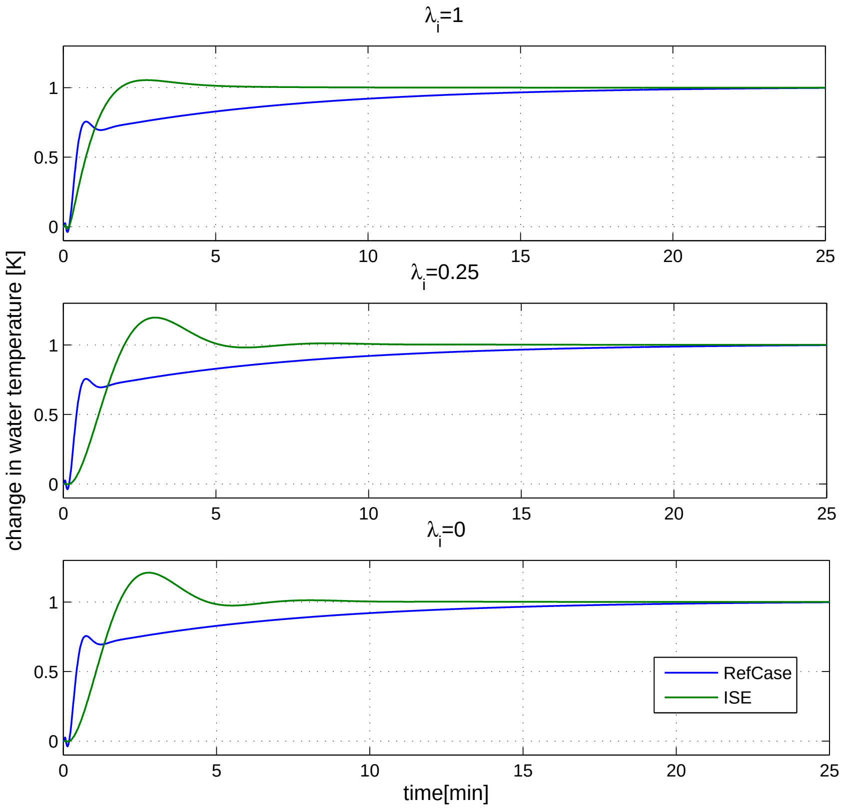

Figure A.5. Step responses of the output y for the reference case as well as for different values of λ_i.

Appendix B

Solution Concepts for bargaining games

When selecting the final solution from the Pareto-optimal set, a decision maker is demanded. The problem of extracting one particular point in a Pareto-optimal set is considered as a bargaining problem. Here, solution concepts for Nash bargaining games are applied as the decision maker and are motivated from their use in control system design.

B.1 The decision maker (DM)

Starting with a given control system, the belonging control-theoretic requirements and constraints, a final solution should be obtained. This is achieved by using the game-theoretic approach. The final solution of the game is selected from the Pareto-optimal set, using a DM that provides a set of controller parameters.
The motivation for a game-theoretic DM has been given in Section 3.4.1.
The solution concepts of the Nash bargaining solution, and the Kalai- Smorodinsky solution are presented for comprehension in the two dimensional case.
The Egalitarian solution is omitted in this case, as the RO example application shows too large differences between the value ranges of J_1 and J_2.

B.1.0.1 The Nash bargaining solution (NB)

The Nash bargaining solution, $NB(\mathcal{PS})$, is obtained by the players simply by maximizing Nash's product. In the two dimensional case, the Nash-bargaining solution is calculated as

$$NB(\mathcal{PS}) = max(J_{cf_1} - (\text{max}J_{cf_1})) \cdot (J_{cf_2} - (\text{max}J_{cf_2})) \tag{B.1}$$

in $\mathcal{PS}$, with d as the disagreement point. According to the notion of John Nash, the Nash bargaining solution includes a fair negotiation resolution, accepted by the rational players. The function fnb, defined through (B.1), assigns to each bargaining game $(\mathcal{PS}, d)$, exactly one cost vector J_{cf}, the Nash solution, and satisfies the following four axioms:

(1) Scale invariance.
The Nash bargaining solution $NB(\mathcal{PS})$ is independent of the units. So, the solution does not vary if the utility is multiplied by a positive constant.

(2) Symmetry.
If $(\mathcal{PS}, d)$ is a symmetric bargaining game, then $fnb_1(\mathcal{PS}, d) = fnb_2(\mathcal{PS}, d)$.

(3) Independence of irrelevant alternatives.
$fnb(\mathcal{PS}, d) = dnb(\mathcal{QS}, d)$ if $(\mathcal{PS}, d)$ and $(\mathcal{QS}, d)$ are bargaining games with an equal disagreement point d, $\mathcal{PS}$ as a subset of $\mathcal{QS}$ and $fnb(\mathcal{QS}, d)$ as an element of $\mathcal{PS}$.

(4) Pareto optimality.
Is $(\mathcal{PS}, d)$ a bargaining game, if $x_1 \geq fnb_1(\mathcal{PS}, d)$ and $x_2 \geq fnb_2(\mathcal{PS}, d)$, then $x \neq fnb(\mathcal{PS}, d)$ in $\mathcal{PS}$.

According to the intention of John Nash, the Nash bargaining solution provides an equitable final solution.

B.1.0.2 Kalai-Smorodinsky solution (KS)

The best known variation of the Nash bargaining solution is the Kalai-Smorodinski solution. Here the third Nash axiom for the Nash bargaining solution is replaced

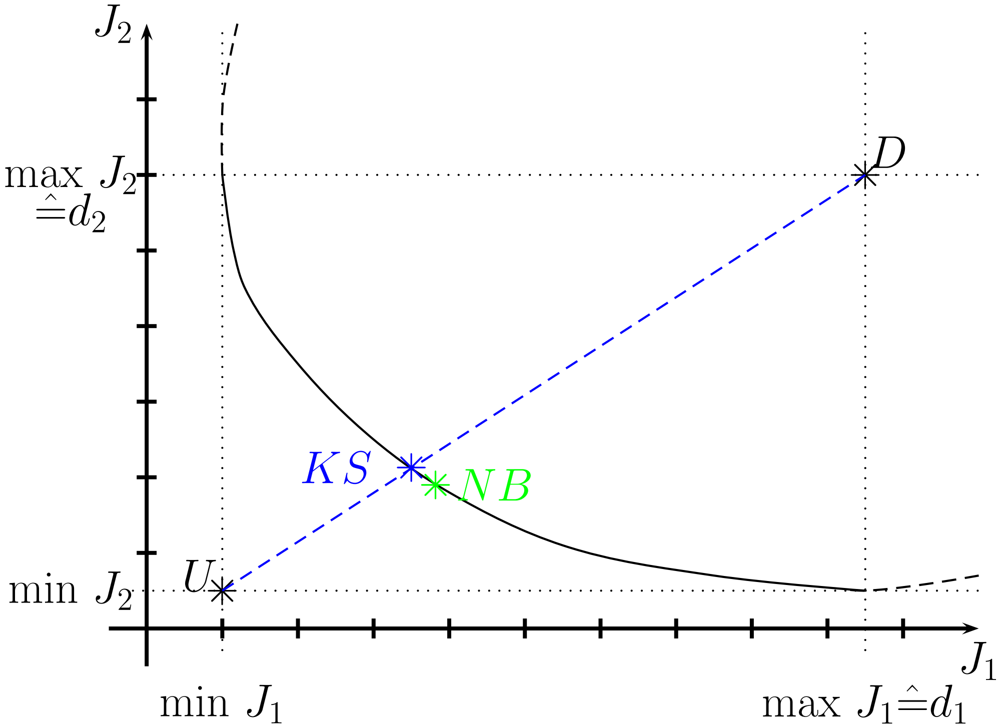

Figure B.1. Solution concepts for bargaining games. Blue: Kalai-Smorodinsky solution, green: Nash bargaining solution.

by the monotonicity axiom:

3') Monotonicity.
If the negotiation set $\mathcal{PS}$ is enlarged, such that the minimum utilities of the players remain unchanged, then neither of the players must not suffer from it.

According to 3′), the Kalai-Smorodinsky solution is situated at the intersection of the Pareto-optimal curve and the straight line linking the disagreement point and the utopia point (Holler u. Illing, 2000). The utopia point is defined as theoretical best point, given through the point $(\min\ J_{cf_1}, \min\ J_{cf_2}, \ldots, \min\ J_{cf_N})$ in the utility space.
A graphical interpretation of the three presented solution concepts for bargaining games is given in Fig.B.1.

B.2 Example

The proposed game-theoretic approach, including the solution concepts are now applied on a reverse osmosis desalination plant as a discrete, cooperative game with two players and the triangular control structure of Fig.B.2.

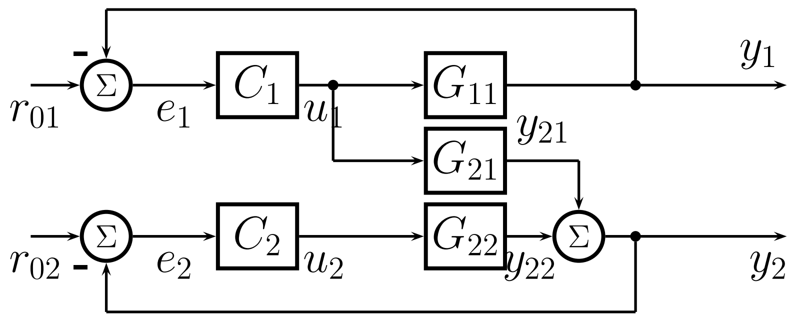

Figure B.2. Multi-loop control structure of the reverse osmosis desalination plant.

Several papers were published, for example (Assef u. a., 1995), (Riverol u. Pilipovik, 2005) and (Robertson u. a., 1996), where RO system identification were considered. The system interaction can be rewritten, compare section 6 for the discrete case with a sample time of $T_0 = 0.2$ as

$$G_{11}(z) = \frac{0.002013z - 2.225 \cdot 10^{-5}}{z^2 - 0.005708z + 0.001273},$$

$$G_{21}(z) = \frac{-0.1574z + 0.08829}{z^2 - 1.383z + 0.5183}$$

and

$$G_{22}(z) = \frac{-6.084z + 3.242}{z^2 - 1.499z + 0.5488}.$$

According to Subsection 2.2.2, the control system design of the reverse osmosis system can now be described as a difference game between two players.

The requirements on the reverse osmosis system are set only on stability and a reference tracking, with minimum deviation as described in Subsection 3.2.

For the control system design of the multi-loop system, the calculation of the error signals E_1 and E_2 for the costs J_{cf_i} in (3.23) can be derived from Fig.B.2 with the step reference signals $r_{01} = z/z - 1$ and $r_{02} = z/z - 1$ as

$$E_{1_{ec}} = \frac{A_{11}z}{A_{11}P_1 + B_{11}Q_1},$$

and

$$E_{2_{ec}} = \frac{(A_{21}A_{22}(A_{11}P_1 + B_{11}Q_1) - B_{21}Q_{11}A_{11}A_{22})z}{A_{21}(A_{11}P_1 + B_{11}Q_1)(A_{22}P_2 + B_{22}Q_2)}.$$

B.2.1 Course and Solution of the game

For the discrete reverse osmosis system, the genetic algorithm operates with 100 generations and 4 chromosomes, two for each controller. Two subpopulations with 500 individuals each are chosen and the number of cost functions is 2.

B.2.1.1 Obtaining a Pareto-optimal set and the final solution

For the example of the reverse osmosis system, several PI-controllers are designed using the proposed approach, where the parameter vector $\boldsymbol{\chi}_{sc}$, with sc, indexing the solution concept, is defined as

$$\boldsymbol{\chi}_{sc} = [q_{01}\ q_{11}\ q_{02}\ q_{12}]^T . \tag{B.2}$$

Obtained Pareto-optimal sets, depending on the different cost functions (3.24),(3.25) and (3.26) are shown in Fig.B.3, Fig.B.4 and Fig.B.5. Here the corresponding Nash bargaining (NB) solutions, the Kalai-Smorodinski (KS) solutions, the disagreement points D, the utopia points U as well as the Pareto-optimal points are highlighted.

The Pareto-optimal fronts vary in their value range due to the incorporation of the elapsed time, represented through the additional factor k and k^2 in the cost function implementations. In contrast, what all three Pareto-optimal fronts have in common is that, the value ranges for the first cost function J_{cf_1} are considerably smaller than the value ranges for the cost functions of J_{cf_2}. This fact is justified through the triangular control structure. Here only the second control loop is disturbed by the first control loop. A change in the set point of r_{01} has effects on the costs J_{cf_1} and J_{cf_2}, while a change in the set point of r_{02} has effects only on the cost J_{cf_2}. According to this, the value range for J_{cf_2} is increased, compared to the value range of J_{cf_1}.
Regarding the final solutions for all cost functions, the KS and the NB solution are placed close to each other, while the E solution increases are small, considering cost

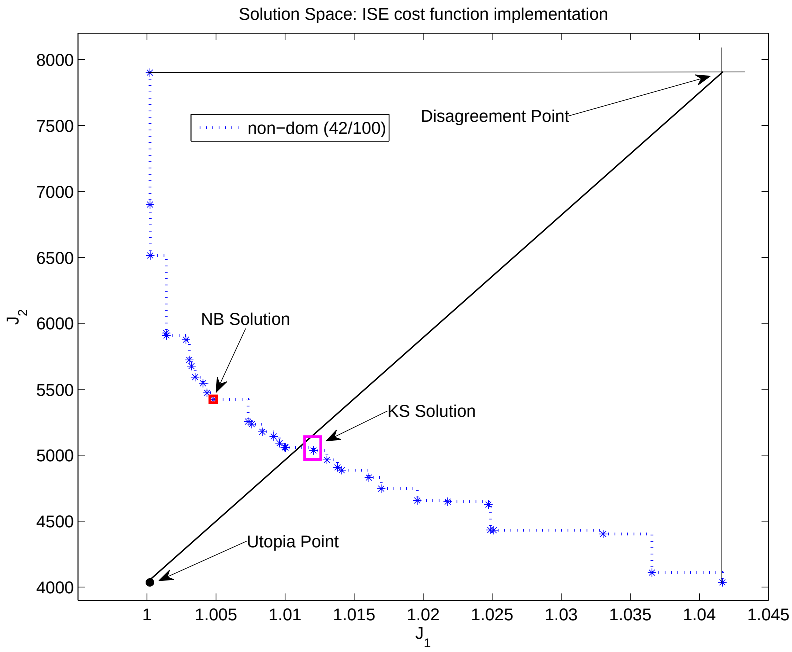

Figure B.3. Solution space for the ISE cost function implementation. Blue stars: Pareto-optimal points.

function J_{cf_1}, but decreases are larger, considering cost function J_{cf_2}, depending on the cost function implementations.

The corresponding controller parameters for each cost function in combination with the three presented solution concepts of (3.24), (3.25), and (3.26) are summarized in Table B.1.

Table B.1. Controller parameters for the reverse osmosis system.

Cost index	Sol. Conc.	q_{01}	q_{11}	q_{02}	q_{12}
ISE	NB	463.3207	-3.7457	-0.3149	0.1741
ISE	KS	443.0338	-1	-0.3336	0.1536
$ITSE$	NB	453.3434	-2.5082	-0.3302	0.1658
$ITSE$	KS	442.8412	-2.2423	-0.3258	0.1584
$ISTSE$	NB	478.4155	-3.1656	-0.3552	0.1590
$ISTSE$	KS	457.7723	-1.0048	-0.3548	0.1581

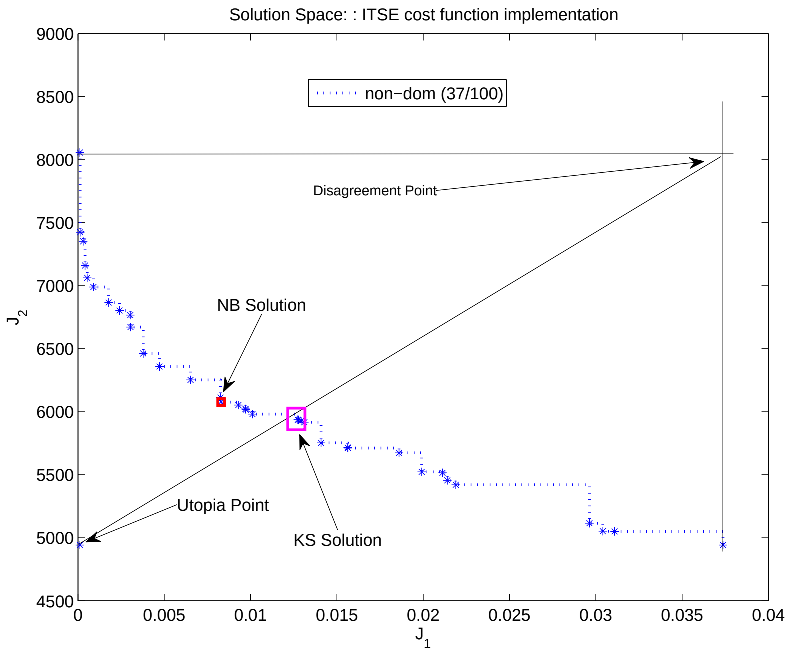

Figure B.4. Solution space for the ITSE cost function implementation. Blue stars: Pareto-optimal points.

B.2.2 Simulation results

To be able to compare the different game solutions of Table B.1, depending on the solution concepts as well as on the formulation of the cost functions, simulation studies are carried out using Matlab/Simulink and corresponding step responses of the reverse osmosis system are analyzed.

A set point change of 0.4 gpm in the permeate flux is performed for all case studies on the system. The corresponding step responses are shown in Fig. B.6.

A set point change of 10 μS/cm in the conductivity is performed for all case studies on the system. The corresponding step responses are shown in Fig. B.7.

Regarding the step responses of Fig.B.6 and Fig.B.7, there is no considerable difference, depending on the solution concept.

However, considering the final solution of KS, the solutions are obtained through the intersection of a determined straight line and the Pareto-optimal set. Also, in the present work, the Pareto-optimal set is defined through a finite number of points with the result that it is possible, that no intersection exist. The alternative

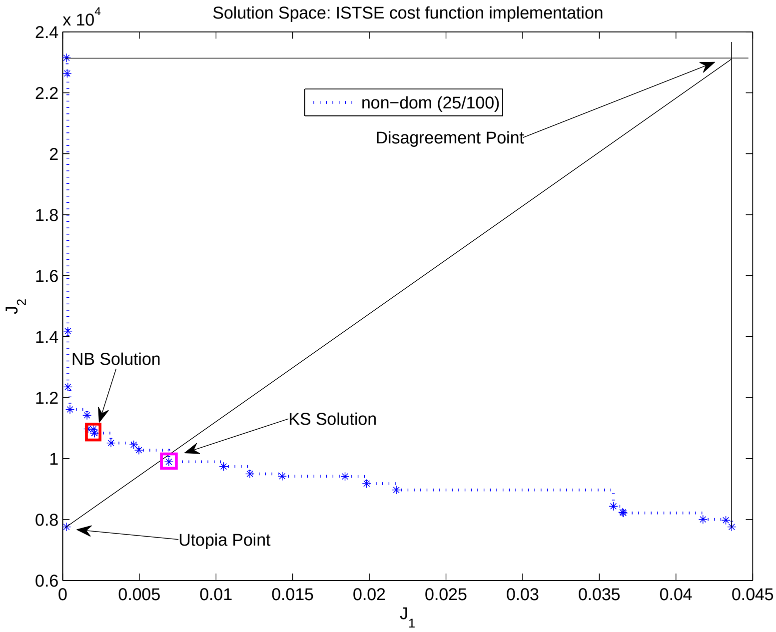

Figure B.5. Solution space for the ISTSE cost function implementation. Blue stars: Pareto-optimal points.

is, to choose the Pareto-optimal point with the shortest distance to the straight line. In addition, the difference of the step responses for the different cost function implementations is small for this example, as well.

Concluding, the authors propose the NB solution as solution concept, due to the computability advantage of the NB solution. Dependent on the Pareto-optimal points, delivered from the genetic algorithm, the NB solution is obtained in minimizing the Nash product.

B.2.3 Conclusion

Using the Nash bargaining solution concept as preliminary decision maker is suggested by the authors. The Nash bargaining solution concept will be used in the further application of the approach, due to the computability property. The Nash bargaining game could be easily extended to the more dimensional case, as well. Regarding the cost indices ISE, ITSE and ISTSE, no preference can be given. Similar simulation results are obtained using the three presented cost function implementations.

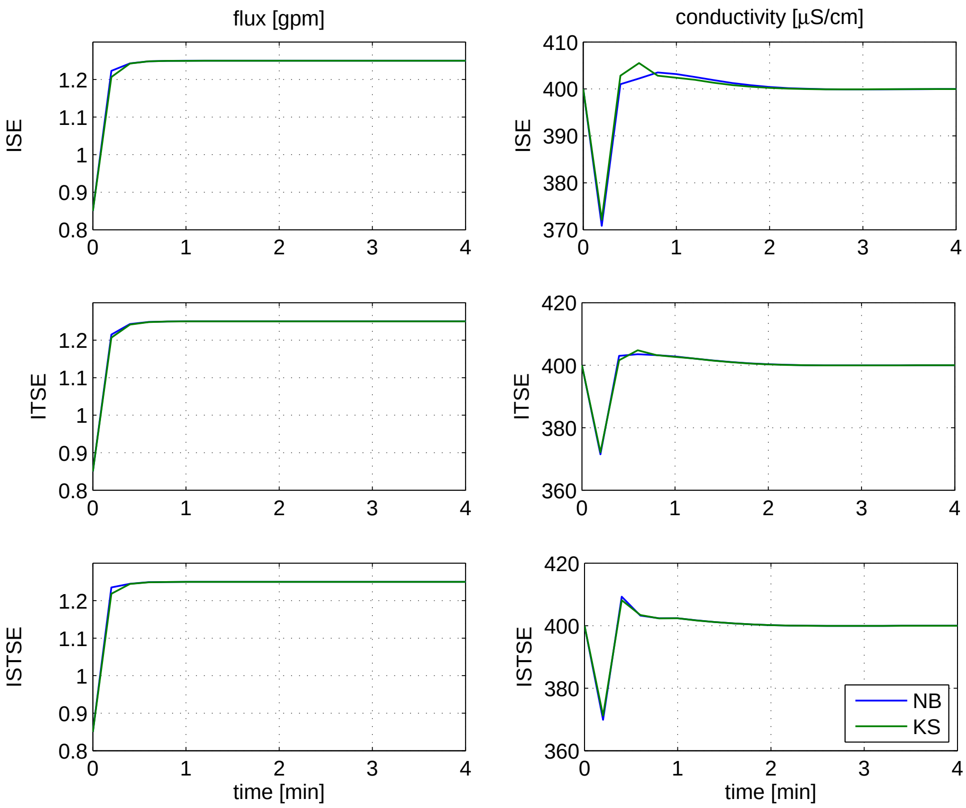

Figure B.6. Step responses to a change in the set point of the permeate flux for the two outputs *flux* and *conductivity* of the reverse osmosis system, depending on the applied solution concepts.

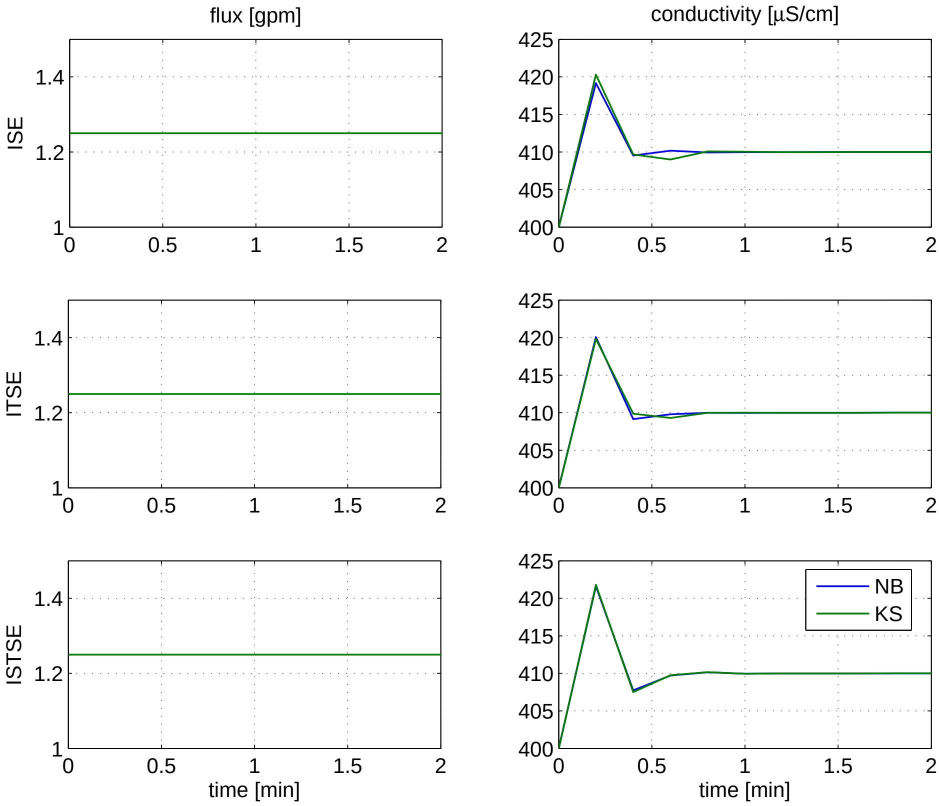

Figure B.7. Step responses to a change in the set point of the conductivity for the two outputs *flux* and *conductivity* of the reverse osmosis system depending on the applied solution concepts.

Appendix C

Equation derivation for the different topologies

The derivation of the error functions E_1 and E_2 for the different game topologies of chapter 4 is given in this chapter.

C.1 Derivation of error equation (4.6)

From Figure 4.1 the following equation for E_1 and Y_1 are derived:

$$E_1 = R_1 - Y_1 \tag{C.1}$$

and

$$Y_1 = G_{11}C_{11}E_1. \tag{C.2}$$

Inserting equation (C.2) in equation (C.1) results in the equation for E_1:

$$E_1 = R_1 - G_{11}C_{11}E_1. \tag{C.3}$$

Solving equation (C.3) to E_1 results in

$$E_1 = \frac{R_1}{(1 + G_{11}C_{11})}.$$

A substitution of G_{11} with $\frac{B_{11}}{A_{11}}$ as well as C_{11} with $\frac{Q_{11}}{P_{11}}$ results after further simplification in the final equation for E_1:

$$E_1 = \frac{A_{11}P_{11}R_1}{A_{11}P_{11} + B_{11}Q_{11}}. \tag{C.4}$$

C.2 Derivation of Error equation (4.7)

From Figure 4.1 the following equations for E_2 and Y_2 are derived:

$$E_2 = R_2 - Y_2 \tag{C.5}$$

and

$$Y_2 = Y_{21} + Y_{22} = G_{21}C_{11}E_1 + G_{22}C_{22}E_2. \tag{C.6}$$

Inserting equation (C.6) in equation (C.5) results in the equation for E_2:

$$E_2 = R_2 - G_{21}C_{11}E_1 - G_{22}C_{22}E_2. \tag{C.7}$$

Solving equation (C.7) to E_2 results in

$$E_2 = \frac{R_2}{1 + G_{22}C_{22}} - \frac{G_{21}C_{11}E_1}{1 + G_{22}C_{22}}. \tag{C.8}$$

A substitution of G_{22} with $\frac{B_{22}}{A_{22}}$, C_{22} with $\frac{Q_{22}}{P_{22}}$, C_{11} with $\frac{Q_{11}}{P_{11}}$ and inserting equation (C.4) in equation (C.8) results after further simplifications in the resolved equation for E_2:

$$E_2 = \frac{A_{21}A_{22}P_{22}(A_{11}P_{11} + B_{11}Q_{11})R_2 - B_{21}Q_{11}A_{11}A_{22}P_{22}R_1}{A_{21}(A_{11}P_{11} + B_{11}Q_{11})(A_{22}P_{22} + B_{22}Q_{22})}. \tag{C.9}$$

C.3 Derivation of error equation (4.9)

From Figure 4.2 the following equations for E_2 and Y_2 are derived:

$$E_2 = R_2 - Y_2 \tag{C.10}$$

and

$$Y_2 = Y_{21} + Y_{22}. \tag{C.11}$$

The equation for Y_{21} of equation (C.11) is given with

$$Y_{21} = G_{21}U_1 = G_{21}C_{11}E_1,$$

resulting after a substitution of G_{21} with $\frac{B_{21}}{A_{21}}$ and C_{11} with $\frac{Q_{11}}{P_{11}}$ in the resolved equation

$$Y_{21} = \frac{B_{21}Q_{11}A11}{A_{21}(A_{11}P_{11} + B_{11}Q_{11})}R_1. \tag{C.12}$$

The equation for Y_{22} of equation (C.11) is given with

$$Y_{22} = G_{22}U_2 \tag{C.13}$$

with

$$U_2 = U_{21} + U_{22}.$$

The resolved equation for U_{21}, substituting C_{21} with $\frac{Q_{21}}{P_{21}}$ and inserting E_1 from equation (C.4) results in

$$U_{21} = C_{21}E_1 = \frac{Q_{21}A_{11}P_{11}}{P_{21}(A_{11}P_{11} - B_{11}Q_{11})}R_1. \tag{C.14}$$

The resolved equation for U_{22} is given as

$$U_{22} = C_{22}E_2. \tag{C.15}$$

Inserting equations (C.14) and (C.15) with additional substitution of C_{22} with $\frac{Q_{22}}{P_{22}}$ in equation (C.13), results after further simplifications in

$$Y_{22} = \frac{B_{22}Q_{21}A_{11}P_{11}}{A_{22}P_{21}(A_{11}P_{11} - B_{11}Q_{11})}R_1 + \frac{B_{22}Q_{22}}{A_{22}P_{22}}E_2. \tag{C.16}$$

Inserting equations (C.16) and (C.12) with (C.11) in equation (C.10), yields to

$$
\begin{aligned}
E_2 =& R_2 \\
&- \frac{B_{21}Q_{11}A_{11}}{A_{21}(A_{11}P_{11} + B_{11}Q_{11})}R_1 - \frac{B_{22}Q_{21}A_{11}P_{11}}{A_{22}P_{21}(A_{11}P_{11} - B_{11}Q_{11})}R_1 \\
&- \frac{B_{22}Q_{22}}{A_{22}P_{22}}E_2
\end{aligned}
$$

and after further simplifications to

$$
\begin{aligned}
E_2 =& \frac{(A_{11}P_{11} + B_{11}Q_{11})A_{21}A_{22}P_{21}P_{22}R_2}{A_{21}P_{21}(A_{22}P_{22} + B_{22}Q_{22})(A_{11}P_{11} + B_{11}Q_{11})} \\
& - \frac{(B_{21}Q_{11}A_{22}P_{21} + B_{22}Q_{21}A_{21}P_{11})A_{11}R_1}{A_{21}P_{21}(A_{22}P_{22} + B_{22}Q_{22})(A_{11}P_{11} + B_{11}Q_{11})}.
\end{aligned} \tag{C.17}
$$

C.4 Derivation of error equations for (4.13) and (4.14)

From Figure 4.3 the following equations for E_1 and Y_1 are derived:

$$E_1 = R_1 - Y_1 \tag{C.18}$$

and

$$Y_1 = G_{11}(C_{11}E_1 + C_{12}E_2). \tag{C.19}$$

Inserting equation (C.19) in (C.18) results in

$$E_1 = \frac{(R_1 - G_{11}C_{12}E_2)}{(1 + G_{11}C_{11})}. \tag{C.20}$$

Regarding E_2, the following equations for E_2 and Y_2 are derived:

$$E_2 = R_2 - Y_2 \tag{C.21}$$

and

$$Y_2 = G_{21}(C_{11}E_1 + C_{12}E_2) + G_{22}C_{22}E_2. \tag{C.22}$$

Inserting equation (C.22) in equation(C.21) results in

$$E_2 = \frac{R_2 - G_{21}C_{11}E_1}{(1 + G_{21}C_{12} + G_{22}C_{22})}. \tag{C.23}$$

Inserting equation (C.23) in equation (C.20) results after further simplifications in the equation for E_1:

$$\begin{aligned} E_1 =& \frac{(1 + G_{21}C_{12} + G_{22}C_{22})}{(1 + G_{11}C_{11})(1 + G_{21}C_{12} + G_{22}C_{22}) - G_{11}C_{12}G_{21}C_{11}} R_1 \\ &- \frac{G_{11}C_{12}}{(1 + G_{11}C_{11})(1 + G_{21}C_{12} + G_{22}C_{22}) - G_{11}C_{12}G_{21}C_{11}} R_2. \end{aligned} \tag{C.24}$$

Regarding E_2, inserting equation (C.20) in equation (C.23) results after fruther simplifications in the equation for E_2:

$$\begin{aligned} E_2 =& \frac{1}{(1 + G_{21}C_{12} + G_{22}C_{22})} \\ & \frac{(1 + G_{21}C_{12} + G_{22}C_{22})(1 + G_{11}C_{11}) - G_{11}C_{12}G_{21}C_{11}}{[(1 + G_{21}C_{12} + G_{22}C_{22})(1 + G_{11}C_{11}) - G_{11}C_{12}G_{21}C_{11}]} R_2 \\ & - (\frac{G_{21}C_{11}(1 + G_{21}C_{12} + G_{22}C_{22})(1 + G_{11}C_{11})}{(1 + G_{21}C_{12} + G_{22}C_{22})(1 + G_{11}C_{11})} \cdot \\ & \frac{1}{[(1 + G_{21}C_{12} + G_{22}C_{22})(1 + G_{11}C_{11}) - G_{11}C_{12}G_{21}C_{11}]}) R_1. \end{aligned} \tag{C.25}$$

Due to manageability, a part of the denominator is substituted with T in the following equation:

$$T = (1 + G_{21}C_{12} + G_{22}C_{22}) - G_{11}C_{12}G_{21}C_{11}.$$

Further simplifications of equation (C.24) and a reordering according to R_1 and R_2 results in:

$$E_1 = \frac{1 + G_{21}C_{12} + G_{22}C_{22}}{T} R_1 - \frac{G_{11}C_{12}}{T} R_2. \tag{C.26}$$

Further simplifications of equation (C.25) and a reordering according to R_1 and R_2 results in:

$$E_2 = \frac{1 + G_{11}C_{11}}{T} R_2 - \frac{G_{21}C_{11}}{T} R_1. \tag{C.27}$$

After substitution of G_{21} with $\frac{B_{21}}{A_{21}}$, G_{22} with $\frac{B_{22}}{A_{22}}$, G_{11} with $\frac{B_{11}}{A_{11}}$, C_{12} with $\frac{Q_{12}}{P_{12}}$, C_{22} with $\frac{Q_{22}}{P_{22}}$, C_{11} with $\frac{Q_{11}}{P_{11}}$ and further simplifying the resulting resolved equations for (C.26) and (C.27), the final equations for E_1 and E_2 are given with:

$$\begin{aligned} E_1 =& \frac{(A_{21}P_{12}A_{22}P_{22} + B_{21}Q_{12}A_{22}P_{22} + B_{22}Q_{22}A_{21}P_{12})A_{11}P_{11}R_1}{T} \\ &- \frac{B_{11}Q_{12}P_{11}A_{21}A_{22}P_{22}R_2}{T} \end{aligned} \tag{C.28}$$

and

$$\begin{aligned} E_2 =& \frac{(A_{11}P_{11} + B_{11}Q_{11})A_{21}P_{12}A_{22}P_{22}R_2}{T} \\ &- \frac{B_{21}Q_{11}A_{11}P_{12}A_{22}P_{22}R_1}{T}. \end{aligned} \tag{C.29}$$

Bibliography

[Allgöwer] Allgöwer, F.: *Mehrgrößenregelung - Handouts zur Vorlesung RT2.* available at http://www.ist.uni-stuttgart.de/education/courses

[Andersson 2000] Andersson, J.: *A survey of multi-objective optimization in engineering design.* Technical Report No. LiTH-IKP-R-1097, Department of Mechanical Engineering, Linköping University, 2000

[Assef u. a. 1995] Assef, J. Z. ; Watters, J.C. ; Desphande, P.B. ; Alatiqi, I.M.: *Advanced Control of a Reverse Osmosis Desalination Unit.* Proceedings of the International Desalination Association (IDA) World Congress, Vol. V, 174-788, Abu Dhabi, 1995

[Aström 1970] Aström, K. J.: *Introduction to Stochastic Control Theory.* London, Academic Press, Inc., 1970

[Balas u. a.] Balas, G. J. ; Doyle, J.C. ; Glover, K. ; Packard, A. ; Smith, R.: *μ-Analysis and Synthesis Toolbox - For Use with MATLAB.* available at *http://www.mathworks.com*, Version 3

[Basar u. Olsder 1999] Basar, T. ; Olsder, G. J.: *Dynamic Noncooperative Game Theory.* Second Edition, Classics in Applied Mathematics, 1999

[Basar u. P.Bernhard 1995] Basar, T. ; P.Bernhard: *H^∞-Optimal Control and Related Minimax Design Problems.* Birkhäuser, Boston, Basel, Berlin, 1995

[Beasley u. a. 1993] Beasley, D. ; Bull, D. R. ; Martin, R. R.: *Dynamic Noncooperative Game Theory.* University Computing, 15(2) pp 58-69, 1993

[Becerra] Becerra, V. M.: *Robustness Analysis for MIMO Systems.* http://www.personal.rdg.ac.uk/ shs99vmb/notes/

[Bernard 2005] Bernard, T.: *Multicriteria optimization of a chemical process with many constraints, (written in german: Multikriterielle Optimierung eines chemischen Prozesses mit vielen Gütekriterien).* GMA-Kongress 2005, VDI-Berichte 1883, pp 393-399, Baden-Baden, 2005

[Bernard u. a.] Bernard, T. ; Sajidman, M. ; Kuntze, H.-B.: *Multicriteria Optimization of complex, dynamic systems with fuzzy decision making, (written in german: Multikriterielle Optimierung von komplexen dynamischen Systemen mit Fuzzy Decision Making).* GOR-Workshop

[Brams 1990] Brams, S. J.: *Negotiation Games: Applying Game Theory to Bargaining and Arbitration.* New York: Routledge, 1990

[Bristol 1966] Bristol, E.H.: *On a new measure of interactions for multivariable control.* IEEE Transactions on Automatic Control, Vol.11, Issue: 1, 1966

[Brosilow u. Joseph 1999] Brosilow, C. ; Joseph, B.: *Techniques of model-based control.* Prentice Hall International Series in the Physical and Chemical Engineering Sciences, 1999

[Calistru 1999] Calistru, C. N.: *Mixed H_2/H_∞ PID Robust Control via Genetic Algorithms and MATHEMATICA Facilities.* The 2nd European Symposium on Intelligent Techniques, Crete, Ortodox Academy of Crete, Kolympari, Chania ESIT'99, 1999

[Clarke u. Gawthrop 1997] Clarke, D. W. ; Gawthrop, P. J.: *On approximate solutions in convex vector optimization.* SIAM Journal on Control and Optimization, 35(6), pp 2128-2136, 1997

[Conley u. Salgado 2000] Conley, A. ; Salgado, M.E.: *Gramian based interaction measure.* Proceedings of the 39th IEEE Conference on Decision and Control, Sydney, Australia, 2000

[Dubey u. Shapley 1979] Dubey, P. ; Shapley, L. S.: *Mathematical properties of the Banzhaf power index.* Mathematics of Operations Research, Vol. 4, No. 2, pp. 99-131, 1979

[Dutta u. Vetrivel 2001] DUTTA, J. ; VETRIVEL, V.: *On approximate minima in vector optimization.* Numerical Functional Analysis and Optimization, 22(7-8), pp 845-859, 2001

[Ehtamo u. Hämäläinen] EHTAMO, H. ; HÄMÄLÄINEN, R. P.: *Decision Making - Negotiation Analysis.* www.negotiation.hut.fi, Helsinki University of Technology

[Elia u. Dahleh 1997] ELIA, N. ; DAHLEH, M.A.: *Controller Design with Multiple Objectives.* IEEE Transactions on Automatic Control, Vol. 42, No.5, 1997

[Erickson u. Hedrick 1999] ERICKSON, K.T. ; HEDRICK, J.L.: *Plantwide Process Control.* New York ; Weinheim : Wiley, 1999

[Farina u. Amato] FARINA, M. ; AMATO, P.: *A Fuzzy Definition of*

[Ferguson] FERGUSON, T. S.: *Game Theory - Introduction.* www.math.ucla.edu/ tom/

[Gambier 2007] GAMBIER, A.: *Multi-objective Optimal Control: An Overview.* Proceedings of the 2007 IEEE Conference on Control Applications, Singapore, 2007

[Gambier u. a. 2006] GAMBIER, A. ; WELLENREUTHER, A. ; BADREDDIN, E.: *Optimal Control of a Reverse Osmosis Desalination Plant Using Multi-objective Optimization.* Proceedings of the IEEE International Conference on Control Applications, Munich, 2006

[Ghavipanjeh 2006] GHAVIPANJEH, F.: *Multivariable Modeling and Control of a Wastewater Benchmark System by PIP Control Design.* Proceedings of the International Control Conference (ICC), Glasgow, Scotland, UK, 2006

[Hart u. Mas-Colell 1997] HART, S. ; MAS-COLELL, A.: *Classical Cooperative Theory I: Core-Like Concepts.* Cooperation: Game Theoretic Approaches, Springer Verlag, pp. 35-42, 1997

[Herreros u. a. 1999] HERREROS, A. ; BAEYENS, E. ; J.R.PERA: *Design of Multi-objective Robust Controllers Using Genetic Algorithms.* Proceedings of the 1999

Genetic and Evolutionary Computation Conference. Workshop Program, pages 131-132, Orlando, Florida, 1999

[Häggblom 1997] Häggblom, K. E.: *Control structure analysis by partial relative gains*. Proceedings of the 36th Conference on Decision and Control, San Diego, California, USA, 1997

[Holland 1992] Holland, J. H.: *Adaptation in natural and artificial systems : an introductory analysis with applications to biology, control, and artificial intelligence*. Cambridge, Mass., MIT Pr., 1992

[Holler u. Illing 2000] Holler, M. J. ; Illing, G.: *Einführung in die Spieltheorie*. 4th ed., Springer-Verlag, Berlin, Heidelberg, New York, 2000

[Hovd u. Skogestad 1992] Hovd, M. ; Skogestad, S.: *Simple frequency-dependent tools for con- trol system analysis, structure selection and design*. Automatica, Vol 28, Issue: 5, 1992

[Hutauruk u. Brown 2005] Hutauruk, N. B. C. ; Brown, M.: *Directed Multi-Objective Optimization for Controller Design*. International Conference on Instrumentation, Communication and Information Technology (ICICI), Bandung, Indonesia, 2005

[Isaacs 1999] Isaacs, R.: *Differential Games - A mathematical theory with applications to warefare and pursuit, control and optimization*. Dover Publications, Inc., Mineola, New York, 1999

[Isermann 1989] Isermann, R.: *Digital Control Systems*. Springer Verlag, 2nd Ed., 1989

[J. Neumann 2004] J. Neumann, O. M.: *Theory of Games and Economic Behavior*. Princeton University Press. 60. anniversary ed., 2004

[Johnson u. Moradi 2005] Johnson, M. A. ; Moradi, M.H.: *PID Control: New Identification and Design Methods*. Springer Verlag London Limited, 2005

[Kawabe u. Tagami 1999] Kawabe, T. ; Tagami, T.: *A New Genetic Algorithm using Pareto Partitioning Method for Robust Partial Model Matching PID*

Design with Two Degrees of Freedom. Proceedings of the Third International ICSC (International Computer Science Conventions) Symposia on Intelligent Industrial Automation (IIA'99) and Soft Computing (SOCO'99), pages 562-567, Genova, 1999

[Konak u. a. 2006] Konak, A. ; Coit, D. W. ; Smith, A. E.: *Multi-objective optimization using genetic algorithms: A tutorial.* Reliability Engineering and System Safety 91, pages 992-1007, 2006

[Kookos u. a. 1999] Kookos, I.K. ; Arvanitis, K.G. ; Kalogeropoulos, G.: *PI Controller Tuning via Multiobjective Optimization.* Proceedings of the 7th Mediterranean Conference on Control and Automation, Haifa, Israel, 1999

[LaValle 2006] LaValle, S. M.: *Planning Algorithms.* Part 3, Decision-Theoretic Planning, available at: http://planning.cs.uiuc.edu/, 2006

[Lemaire 1991] Lemaire, J.: *Cooperative Game Theory and its Insurance Applications.* Invited Paper, Astin Bulletin, Vol.21, No.1, pp. 17-40, 1991

[Lin 1976] Lin, J.G.: *MultiObjective Problems: Pareto-Optimal Solutions by Method of Proper Equality Constraints.* IEEE Transactions on Automatic Control, Vol. AC-21, No.5, 1976

[Liu u. a. 2002] Liu, G. P. ; Yang, J. B. ; Whidborne, J. F.: *Multiobjective Optimization and Control.* Research Studies Press LTD., 2002

[Luce u. Raiffa 1989] Luce, R. D. ; Raiffa, H.: *Games and Decisions: Introduction and Critical Survey.* Dover Publications, 1989

[Lunze 2004] Lunze, J.: *Regelungstechnik 2.* Springer, Berlin, 2004

[Lygeros u. a. 1995] Lygeros, J. ; Godbole, D. N. ; Sastry, S.: *A Game Theoretic Approach to Hybrid System Design.* Technical Report UCB/ERL-M95/77, available at: http://citeseer.ist.psu.edu/lygeros95game.html, 1995

[Lygeros u. a. 1996] Lygeros, J. ; Godbole, D. N. ; Sastry, S.: *Multiagent Hybrid System Design using Game Theory and Optimal Control.* Proceedings of the IEEE Conference on Decision and Control, pp. 1190–1195, Kobe, Japan, 1996

[Lygeros u. a. 1997] Lygeros, J. ; Godbole, D. N. ; Sastry, S.: *A Design Framework for Hierarchical, Hybrid Control.* California Partners for Advanced Transit and Highways (PATH). Research Reports: Paper UCB-ITS-PRR-97-24, available at: http://repositories.cdlib.org/its/path/reports/UCB-ITS-PRR-97-24, 1997

[Makowski 1994] Makowski, M.: *Methodology and a modular tool for multiple criteria analysis of lp models.* Technical Report WP-94-102, International Institute for Applied Systems Analysis, 1994

[Manoso u. a. 1997] Manoso, C. ; Hernandez, R. ; Madrid, A. P. ; Dormito, S.: *Robust Stability Analysis of GPC: An application to dead-beat and mean-level predictive controllers.* 5th Mediterranean Conference on Control and Systems, Paphos, Cyprus, 1997

[Marler u. Arora 2004] Marler, R. T. ; Arora, J.S.: *Survey of Multi-objective Optimization Methods for Engineering.* Structural and Multidisciplinary Optimization, 26, 6, 369-395, 2004

[McCain] McCain, R. A.: *Cooperative Games.* available at: http://williamking.www.drexel.edu

[Michalewicz 1995] Michalewicz, Z.: *A Survey of Constraint Handling Techniques in Evolutionary Computation Methods.* Proceedings of the 4th Annual Conference on Evolutionary Programming, MIT Press, Cambridge, MA, pp. 135-155, 1995

[Myerson 1991] Myerson, R. B.: *Game Theory - Analysis of conflict.* Harvard University Pr., Cambridge, Massachusetts, 1991

[Natto 2007] Natto, S.: *Entwurf von Regelsystemen auf Basis multikriterieller Optimierung für eine verfahrenstechnische Anlage.* Diplomarbeit, LS Automation, Central Institute of Technical Informatics, University of Heidelberg, Germany, 2007

[Osborne u. Rubinstein 2001] Osborne, M. J. ; Rubinstein, A.: *A course in game theory*. seventh printing, MIT Press, Cambridge MA, 2001

[Osyczka 1985] Osyczka, A.: *Multicriteria optimization for engineering design*. in Design Optimization, J. S. Gero, Ed. New York: Academic, pp. 193–227, 1985

[Pohlheim 2000] Pohlheim, H.: *Evolutionäre Algorithmen - Verfahren, Operatoren und Hinweise für die Praxis*. Springer Verlag, Heidelberg, 2000

[Pohlheim 2001] Pohlheim, H.: *Evolutionary Algorithms: Overview, Methods and Operators*. available at *http://www.geatbx.com*, 2001

[Qiang Xiong u. He] Qiang Xiong, Wen-Jian C. ; He, Ming

[Rhee u. Speyer 1989] Rhee, I. ; Speyer, J. L.: *A Game Theoretic Controller and its Relationship to H^{∞} and Linear-Exponential-Gaussian Synthesis*. 1989

[Riecks 2006] Riecks, C.: *Was ist ein Spiel?* Grundlagen der Spieltheorie, http://www.spieltheorie.de/, 2006

[Riverol u. Pilipovik 2005] Riverol, C. ; Pilipovik, V.: *Mathematical Modeling of a perfect decoupled control system and its application: A reverse osmosis desalination industrial-scale unit*. Journal of Automated Methods and Management in Chemistry, 2005

[Robertson u. a. 1996] Robertson, M. W. ; Watters, J. C. ; Desphande, P.B. ; Assef, J.Z. ; Alatiqi, I.M.: *Model based control for reverse osmosis desalination processes*. Desalination, 104, 59-68, 1996

[Rosenbrock 1974] Rosenbrock, H. H.: *Computer-aided Control System Design*. Academic Press, 1974

[Saksala 2004] Saksala, T.: *Nash Equilibrium in Bicriteria Structural Optimization*. Proceedings of the XXI International Congress of Theoretical and Applied Mathematics (ICTAM), Warsaw, Poland, 2004

[Salgado u. Conlea 2004] Salgado, M.E. ; Conlea, A.: *MIMO interaction measure and controller structure selection*. International Journal of Control, Vol. 7, Issue: 4, 2004

[Skogestad 2003] Skogestad, S.: *Control structure design: What should we control, measure and manipulate?* Proceedings of the first African Control Conference (Afcon), University of Cape Town, South Africa, 2003

[Skogestad u. Postlethwaite 1996] Skogestad, S. ; Postlethwaite, I.: *Multivariable Feedback Control - Analysis and Design*. John Wiley & Sons; England, 1996

[Tagami u. a. 2004] Tagami, T. ; Kawabe, T. ; Ikeda, K.: *Multi-objective Design Scheme for Robust I-PD Controller*. Proceedings of IASTED International Conference on Modelling, Identification and Control (MIC 2004), pp. 128-131, Grindelwald, Switzerland, 2004

[Tomlin u. a. 2000] Tomlin, C. ; Lygeros, J. ; Sastry, S.: *A Game Theoretic Approach to Controller Design for Hybrid Systems*. Proceedings of the IEEE, Vol. 88, Issue 7, pp. 949-970, 2000

[Turocy u. v. Stengel 2001] Turocy, T. L. ; Stengel, B. v.: *Game Theory*. CDAM Research Report LSE-CDAM-2001-09, downloadable at: http://www.cdam.lse.ac.uk/, 2001

[Vincent u. Leitmann 1970] Vincent, T. L. ; Leitmann, G.: *Control-Space Properties of Cooperative Games*. Journal Of Optimization Theory and Applications: Vol. 6, No. 2, 1970

[de Weck 2004] Weck, O.L. de: *Multiobjective Optimization: History and Promise*. Invited Keynote Paper, GL2-2, The Third China-Japan-Korea Joint Symposium on Optimization of Structural and Mechanical Systems, Kanazawa, Japan, 2004

[Wellenreuther u. a. 2006a] Wellenreuther, A. ; Gambier, A. ; Badreddin, E.: *Multi-loop Controller Design for a Heat Exchanger*. Proceedings of the IEEE International Conference on Control Applications, Munich, 2006

[Wellenreuther u. a. 2006b] Wellenreuther, A. ; Gambier, A. ; Badreddin, E.: *Parameter tuning of multiple controllers for multiloop systems using game theory*. Proceedings of the 6th Asian Control Conference, Bali, Indonesia, 2006

[Wellenreuther u. a. 2007] Wellenreuther, A. ; Gambier, A. ; Badreddin, E.: *Optimal Multi-loop Control System Design Subject to Explicit Constraints.* Proceedings of the European Control Conference, Kos, Greece, 2007

[Wellenreuther u. a. 2008a] Wellenreuther, A. ; Gambier, A. ; Badreddin, E.: *Controller Tuning of Stochastic Disturbed Multi-loop Control Systems.* Proceedings of the IEEE Multi-conference on Systems and Control, San Antonio, Texas, 2008

[Wellenreuther u. a. 2008b] Wellenreuther, A. ; Gambier, A. ; Badreddin, E.: *Game theoretical Analysis for Choosing the Topology in Multi-loop Control Systems.* Proceedings of the American Control Conference, Seattle, Washington, 2008

[Wellenreuther u. a. 2008c] Wellenreuther, A. ; Gambier, A. ; Badreddin, E.: *Game-theoretical Analysis of a Multi-loop Control Structure With Constrained Strategy Sets.* Proceedings of the IEEE Multi-conference on Systems and Control, San Antonio, Texas, 2008

[Zames 1981] Zames, G.: *Feedback and Optimal Sensitivity: Model Reference Transformations, Multiplicative Seminorms, and Approximate Inverses.* 1981. – 301–320 S.